STIC Search Report
EIC 2100

STIC Database Tracking Number: 132622

TO: Merilyn Nguyen **Location: 4B41** **Art Unit : 2171** **Wednesday, September 22, 2004** **Case Serial Number: 09/819022**	**From: Geoffrey St. Leger** **Location: EIC 2100** **PK2-4B30** **Phone: 308-7800** **geoffrey.stleger@uspto.gov**

Search Notes

Dear Examiner Nguyen,

Attached please find the results of your search request for application 09/819022. I searched Dialog's foreign patent files, technical databases, product announcement files and general files.

Please let me know if you have any questions.

Regards,

Geoffrey St. Leger
4B30/308-7800

STIC Search Results Feedback Form

EIC 2100

Questions about the scope or the results of the search? Contact ***the EIC searcher or contact:***

Anne Hendrickson, EIC 2100 Team Leader
308-7831, CPK2-4B40

Voluntary Results Feedback Form

- *I am an examiner in Workgroup:* ☐ *Example: 2133*
- *Relevant prior art **found**, search results used as follows:*
  - ☐ 102 rejection
  - ☐ 103 rejection
  - ☐ Cited as being of interest.
  - ☐ Helped examiner better understand the invention.
  - ☐ Helped examiner better understand the state of the art in their technology.

Types of relevant prior art found:

- ☐ Foreign Patent(s)
- ☐ Non-Patent Literature
  (journal articles, conference proceedings, new product announcements etc.)

- *Relevant prior art **not found:***
  - ☐ Results verified the lack of relevant prior art (helped determine patentability).
  - ☐ Results were not useful in determining patentability or understanding the invention.

Comments:

Drop off or send completed forms to STIC/EIC2100 CPK2-4B40

Access DB# 121602

SEARCH REQUEST FORM

Scientific and Technical Information Center

68

Requester's Full Name: MERLYN NGUYEN Examiner #: 79389 Date: 09/15/04
Art Unit: 2171 Phone Number 305 5177 Serial Number: 09/179022
Mail Box and Bldg/Room Location: 4B41 Results Format Preferred (circle): PAPER DISK E-MAIL

If more than one search is submitted, please prioritize searches in order of need.

****

Please provide a detailed statement of the search topic, and describe as specifically as possible the subject matter to be searched. Include the elected species or structures, keywords, synonyms, acronyms, and registry numbers, and combine with the concept or utility of the invention. Define any terms that may have a special meaning. Give examples or relevant citations, authors, etc, if known. Please attach a copy of the cover sheet, pertinent claims, and abstract.

Title of Invention: Abstract for compression of nodes in a trie structure

Inventors (please provide full names): Livonen, Jukka-Pekka
Tikkanen, Matti

Earliest Priority Filing Date: 09/29/1998

For Sequence Searches Only *Please include all pertinent information (parent, child, divisional, or issued patent numbers) along with the appropriate serial number.*

Memory is implemented in a directory structure comprising a tree shaped hierarchy having nodes at several different levels, nodes including a trie node contains pointer pointing to a lower node. A trie node associated with a logical table, the number of elements in the table corresponding to a power of TWO or a bucket containing at least one element in such a way that the type of an individual element in the bucket is selected from a group including a data unit, a pointer to a stored data unit, a pointer to another directory structure and another directory structure.

KeyWord: directory structure, logical table, bucket, pointer, tree-shaped hierarchy, trie node, bit, memory.

****

STAFF USE ONLY

Searcher: Geoffrey St. Leger
Searcher Phone #: 308-7800
Searcher Location: 4B30
Date Searcher Picked Up: 9/21/04
Date Completed: 9/22/04
Searcher Prep & Review Time: 7
Clerical Prep Time:
Online Time: 220

Type of Search		Vendors and cost where applicable	
NA Sequence (#)		STN	
AA Sequence (#)		Dialog	✓
Structure (#)		Questel/Orbit	
Bibliographic	✓	Dr.Link	
Litigation		Lexis/Nexis	
Fulltext	✓	Sequence Systems	
Patent Family		WWW/Internet	
Other		Other (specify)	

PTO-1590 (8-01)

16/5/12 **(Item 12 from file: 347)**
DIALOG(R)File 347:JAPIO

00710857 **Image available**
CONSTITUTION OF FILE

PUB. NO.: 56-031157 [JP 56031157 A]
PUBLISHED: March 28, 1981 (**19810328)**
INVENTOR(s): SEKI KOHEI
APPLICANT(s): HITACHI LTD [000510] (A Japanese Company or Corporation), JP (Japan)
HITACHI ENG CO LTD [323361] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 54-106044 [JP 79106044]
FILED: August 22, 1979 (19790822)
INTL CLASS: [3] G06F-013/04; G06F-007/22; G06F-015/40
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- **Memory** Units); 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units); 45.4 (INFORMATION PROCESSING -- Computer Applications
JOURNAL: Section: P, Section No. 64, Vol. 05, No. 84, Pg. 95, June 02, 1981 (19810602)

ABSTRACT

PURPOSE: To reduce not only the input and output time but the time for the constitution process of the file, by performing the constitution process of the formation file in such way that the input and the output may always be processes in order to the mobile **memory** device through which the file is produced.

CONSTITUTION: The records of the data region are supplied successively, and only the key is delivered successively to the different work file to produce the file of only the key. Then the **directory** region is produced successively from its head and with every **bucket** . The head key of the work file is supplied and then convêrted into the **bucket address** through the hash method. And in the case of the **address** of the **bucket** 1, the key and its record **address** are produced on the buffer in the form of the record of the **directory** region. In such way, the processes are carried out successively. Then the **bucket** 1 produced on the buffer is delivered to the **directory** region. The above process is given successively for the **buckets** 2 and 3 to produce the **directory** region.

16/5/13 **(Item 1 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

012866490 **Image available**
WPI Acc No: 2000-038323/200003
XRPX Acc No: N00-028925
Memory **for storing** hierarchical **pattern values for use during program execution**
Patent Assignee: MICROSOFT CORP (MICT)
Inventor: BAR O; BERNET Y; DOUCEUR J R
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5995971	A	19991130	US 97933477	A	19970918	200003	B

Priority Applications (No Type Date): US 97933477 A 19970918
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5995971	A		54	G06F-017/30	

Abstract (Basic): US 5995971 A
NOVELTY - Search and pattern structures formed on branch and

pattern nodes, define a **memory** structure. The branch nodes collectively defines a binary search **trie** which indexes into a subset of pattern nodes. The **memory** structure for each branch node, has an address stored to a consecutive node on the path. The **memory** structure for each pattern node, stores corresponding pattern values with respective wild cards.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(a) computer readable storage medium;
(b) input key using method;
(c) input key using apparatus

USE - For use during program execution.

ADVANTAGE - Since rhizome is frequently used with large classification database, significant time saving is enabled, when compared to the conventional classification techniques. In the absence of any wild card based patterns, the rhizome does not engender any performance penalty over use of a conventional Patricia **tree** . The structures can be easily stored in any computer readable media such as floppy disc, magnetic disk, optical disk etc.

DESCRIPTION OF DRAWING(S) - The figure shows flowchart of node insert routine executed by insertion routine.

pp; 54 DwgNo 16A,16B/20

Title Terms: **MEMORY** ; STORAGE; **HIERARCHY** ; PATTERN; VALUE; PROGRAM; EXECUTE
Derwent Class: T01
International Patent Class (Main): G06F-017/30
File Segment: EPI

16/5/15 (Item 3 from file: 350)
DIALOG(R)File 350:Derwent WPIX

012104539 **Image available**
WPI Acc No: 1998-521451/ **199844**
XRPX Acc No: N98-407214
Associative memory **implementation method using digital** tree **structure - in which nodes in digital** tree hierarchy **are compressed to provide single path downward in** tree **-shaped** hierarchy
Patent Assignee: NOKIA TELECOM OY (OYNO); NOKIA NETWORKS OY (OYNO)
Inventor: IIVONEN J; TIKKANEN M
Number of Countries: 081 Number of Patents: 006
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9841933	A1	19980924	WO 98FI192	A	19980304	199844	B
FI 9701067	A	19980915	FI 971067	A	19970314	199849	
FI 102426	B1	19981130	FI 971067	A	19970314	199902	
AU 9866240	A	19981012	AU 9866240	A	19980304	199907	
EP 976066	A1	20000202	EP 98908123	A	19980304	200011	
			WO 98FI192	A	19980304		
US 6505206	B1	20030107	WO 98FI192	A	19980304	200306	
			US 99389574	A	19990903		

Priority Applications (No Type Date): FI 971067 A 19970314
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 9841933	A1	E	26	G06F-017/30	

Designated States (National): AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM GW HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US UZ VN YU ZW
Designated States (Regional): AT BE CH DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG ZW

FI 9701067	A		G06F-017/30	
FI 102426	B1		G06F-017/30	Previous Publ. patent FI 9701067
AU 9866240	A		G06F-017/30	Based on patent WO 9841933
EP 976066	A1	E	G06F-017/30	Based on patent WO 9841933

Designated States (Regional): DE FR GB

US 6505206 B1 G06F-017/30 Cont of application WO 98FI192

Abstract (Basic): WO 9841933 A

The **memory** implementation method involves using a **directory** structure comprising of a **tree** -shaped **hierarchy** having nodes at several different levels. An individual node can be a node comprising an array, in which an individual element may contain the address of a lower node in the **tree** -shaped **hierarchy** . An individual element may also be empty, and the number of elements in the array corresponding to a power of two.

An individual node can also be a **bucket** containing at least one element, such that the type of element in the **bucket** may be a **data** unit, a **pointer** to a stored **data** unit, a pointer to a node in another **directory** structure and another **directory** structure.

USE - In central **memory** databases, and in conjunction with **memories** based on digital **tree** structure.

ADVANTAGE - Enables number of **memory** references requiring computation time to be minimised, improving speed of **memory** operation.

Dwg.7/10

Title Terms: ASSOCIATE; **MEMORY** ; IMPLEMENT; METHOD; DIGITAL; **TREE** ; STRUCTURE; NODE; DIGITAL; **TREE** ; **HIERARCHY** ; COMPRESS; SINGLE; PATH; DOWN; **TREE** ; SHAPE; **HIERARCHY**

Derwent Class: T01

International Patent Class (Main): G06F-017/30

File Segment: EPI

16/5/16 (Item 4 from file: 350)

DIALOG(R)File 350:Derwent WPIX

012104538 **Image available**

WPI Acc No: 1998-521450/ **199844**

XRPX Acc No: N98-407213

Associative memory **implementation method using digital** tree **structure - in which nodes in digital** tree hierarchy **are compressed to provide single path downward in** tree **-shaped** hierarchy

Patent Assignee: NOKIA TELECOM OY (OYNO); NOKIA NETWORKS OY (OYNO)

Inventor: IIVONEN J; TIKKANEN M

Number of Countries: 081 Number of Patents: 006

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9841932	A1	19980924	WO 98FI191	A	19980304	199844	B
FI 9701066	A	19980915	FI 971066	A	19970314	199849	
FI 102425	B1	19981130	FI 971066	A	19970314	199902	
AU 9866239	A	19981012	AU 9866239	A	19980304	199907	
EP 970430	A1	20000112	EP 98908122	A	19980304	200008	
			WO 98FI191	A	19980304		
US 6115716	A	20000905	WO 98FI191	A	19980304	200044	
			US 99389498	A	19990903		

Priority Applications (No Type Date): FI 971066 A 19970314

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 9841932	A1	E	26	G06F-017/30	

Designated States (National): AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM GW HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US UZ VN YU ZW

Designated States (Regional): AT BE CH DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG ZW

FI 9701066	A		G06F-017/30	
FI 102425	B1		G06F-017/30	Previous Publ. patent FI 9701066
AU 9866239	A		G06F-017/30	Based on patent WO 9841932
EP 970430	A1	E	G06F-017/30	Based on patent WO 9841932

Designated States (Regional): DE FR GB

US 6115716 A G06F-017/30 Cont of application WO 98FI191

Abstract (Basic): WO 9841932 A

The **memory** implementation method involves using a **directory** structure comprising of a **tree** -shaped **hierarchy** having nodes at several different levels. An individual node can be a node comprising an array, in which an individual element may contain the address of a lower node in the **tree** -shaped **hierarchy** . An individual element may also be empty.

A **bucket** contains at least one element, such that the type of element in the **bucket** may be a **data** unit, a **pointer** to a stored **data** unit, a pointer to another **directory** structure and another **directory** structure.

USE - In central **memory** databases including database in which large number of insertions and deletions compared to number of retrievals e.g. visitor location registers in mobile communications networks.

ADVANTAGE - Enables number of **memory** references requiring computation time to be minimised, improving speed of **memory** operation.

Dwg.7/10

Title Terms: ASSOCIATE; **MEMORY** ; IMPLEMENT; METHOD; DIGITAL; **TREE** ; STRUCTURE; NODE; DIGITAL; **TREE** ; **HIERARCHY** ; COMPRESS; SINGLE; PATH; DOWN; **TREE** ; SHAPE; **HIERARCHY**

Derwent Class: T01

International Patent Class (Main): G06F-017/30

File Segment: EPI

16/5/17 (Item 5 from file: 350)

DIALOG(R)File 350:Derwent WPIX

012104537 **Image available**

WPI Acc No: 1998-521449/ **199844**

XRPX Acc No: N98-407212

Associative memory **implementation method using digital** tree **structure - in which nodes in digital** tree hierarchy **are compressed to provide single path downward in** tree **-shaped** hierarchy

Patent Assignee: NOKIA NETWORKS OY (OYNO); NOKIA TELECOM OY (OYNO)

Inventor: IIVONEN J; TIKKANEN M

Number of Countries: 081 Number of Patents: 006

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9841931	A1	19980924	WO 98FI190	A	19980304	199844	B
FI 9701065	A	19980915	FI 971065	A	19970314	199849	
FI 102424	B1	19981130	FI 971065	A	19970314	199902	
AU 9866238	A	19981012	AU 9866238	A	19980304	199907	
EP 1008063	A1	20000614	EP 98908121	A	19980304	200033	
			WO 98FI190	A	19980304		
US 6499032	B1	20021224	WO 98FI190	A	19980304	200303	
			US 99390526	A	19990903		

Priority Applications (No Type Date): FI 971065 A 19970314

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 9841931	A1	E	32	G06F-017/30	

Designated States (National): AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM GW HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US UZ VN YU ZW

Designated States (Regional): AT BE CH DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG ZW

FI 102424	B1			Previous Publ. patent FI 9701065
AU 9866238	A			Based on patent WO 9841931
EP 1008063	A1	E	G06F-017/30	Based on patent WO 9841931

Designated States (Regional): DE FR GB

US 6499032	B1		G06F-017/30	Cont of application WO 98FI190

Abstract (Basic): WO 9841931 A

The **memory** implementation method involves using a **directory** structure comprising of a **tree** -shaped **hierarchy** having nodes at several different levels. An individual node can be a node comprising an array, in which an individual element may contain the address of a lower node in the **tree** -shaped **hierarchy** , and an individual element may also be empty. The number of elements in the array corresponds to a power of two.

A **bucket** contains at least one element, such that the type of individual element in the **bucket** is selected from a group including; a **data** unit, a **pointer** to a stored data unit, a pointer to another **directory** structure and another **directory** structure.

USE - In central **memory** databases including database in which large number of insertions and deletions compared to number of retrievals e.g. visitor location registers in mobile communications networks.

ADVANTAGE - Enables number of **memory** references requiring computation time to be minimised, improving speed of **memory** operation.

Dwg.7/10

Title Terms: ASSOCIATE; **MEMORY** ; IMPLEMENT; METHOD; DIGITAL; **TREE** ; STRUCTURE; NODE; DIGITAL; **TREE** ; **HIERARCHY** ; COMPRESS; SINGLE; PATH; DOWN; **TREE** ; SHAPE; **HIERARCHY**

Derwent Class: T01

International Patent Class (Main): G06F-017/30

File Segment: EPI

16/5/18 (Item 6 from file: 350)

DIALOG(R)File 350:Derwent WPIX

011946203 **Image available**

WPI Acc No: 1998-363113/ **199831**

Related WPI Acc No: 1998-363111; 1998-363145; 2000-195798

XRPX Acc No: N98-283490

Key look up method for network packet route determination - involves searching destination tree **by comparing bits in packet destination until matching or near matching destination is found**

Patent Assignee: JUNIPER NETWORKS (JUNI-N)

Inventor: FERGUSON D C; PATEL R N; SINDHU P S; ANAND R K; LIENCRES B O

Number of Countries: 020 Number of Patents: 005

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9827662	A2	19980625	WO 97US23287	A	19971216	199831	B
EP 948849	A2	19991013	EP 97951733	A	19971216	199947	
			WO 97US23287	A	19971216		
JP 2001509978	W	20010724	WO 97US23287	A	19971216	200147	
			JP 98527947	A	19971216		
CA 2274962	C	20020806	CA 2274962	A	19971216	200260	
			WO 97US23287	A	19971216		
JP 3453148	B2	20031006	WO 97US23287	A	19971216	200366	
			JP 98527947	A	19971216		

Priority Applications (No Type Date): US 97901061 A 19970724; US 96767576 A 19961216; US 97844171 A 19970418

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 9827662	A2	E	61	H04B-000/00	
Designated States (National): CA JP					
Designated States (Regional): AT BE CH DE DK ES FI FR GB GR IE IT LU MC NL PT SE					
EP 948849	A2	E		H04J-003/02	Based on patent WO 9827662
Designated States (Regional): CH DE ES FI FR GB IT LI SE					
JP 2001509978	W		60	H04L-012/56	Based on patent WO 9827662
CA 2274962	C	E		H04J-003/02	Based on patent WO 9827662
JP 3453148	B2		28	H04L-012/56	Previous Publ. patent JP 200109978
					Based on patent WO 9827662

Abstract (Basic): WO 9827662 A

The method involves forward traversing of one or more nodes which make a **trie** stored in a **memory** , upon receipt of a key. This is done by evaluating a bit in the key, at each node traversed, as indicated by a bit-to-test indicator associated with each node. A value of the bit in the key determines the path traversed along the **trie** .

An end node is located in the **trie** , in which the end node has a route. The route is compared to the key, and if matching occurs then destination information associated with the end node is output, to guide the transfer of the packet through the routing device. For no matching, the **trie** is traversed backwards to locate a best match for the key.

ADVANTAGE - Provides single unified method of finding best match look-ups for different routing methods and can hold extra data

Title Terms: KEY; UP; METHOD; NETWORK; PACKET; ROUTE; DETERMINE; SEARCH; DESTINATION; **TREE** ; COMPARE; BIT; PACKET; DESTINATION; MATCH; MATCH; DESTINATION; FOUND

Derwent Class: W01

International Patent Class (Main): H04B-000/00; H04J-003/02; H04L-012/56

File Segment: EPI

16/5/20 (Item 8 from file: 350)

DIALOG(R)File 350:Derwent WPIX

010866691 **Image available**

WPI Acc No: 1996-363642/ **199637**

XRPX Acc No: N96-306560

Computer memory **storage method using genetic algorithm for optimising** memory **locations - involves initial population size, with number of generations selected. With increasing population size there is convergence of this algorithm in small number of generations**

Patent Assignee: HEWLETT-PACKARD CO (HEWP)

Inventor: KONSELLA S

Number of Countries: 003 Number of Patents: 004

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
DE 19532371	A1	19960808	DE 1032371	A	19950901	199637	B
JP 8272659	A	19961018	JP 95334560	A	19951222	199701	
US 5651099	A	19970722	US 95378329	A	19950126	199735	
DE 19532371	C2	19971002	DE 1032371	A	19950901	199743	

Priority Applications (No Type Date): US 95378329 A 19950126

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
DE 19532371	A1		19	H03M-007/30	
JP 8272659	A		14	G06F-012/00	
US 5651099	A		20	G06F-015/18	
DE 19532371	C2		10	H03M-007/30	

Abstract (Basic): DE 19532371 A

The original data is stored in a **memory** and then a number of generations is selected (1001), shown in the figure of the data flow. A number of individuals are initialised (1002), and the **tree** size determined (1003) and a population of pairs selected (1007), the probability of selection being proportional to the individual **tree** size. There is also an cross-over operation (1008).

Then, there is an iteration (1004) of the **tree** size determination, the selection (1007) and the operation (1008), with the **tree** of minimum size stored in the **memory** .

USE/ADVANTAGE - Suitable for data processing in DNA analysis. Obtains efficiently **tree** of minimum size in genetic analysis.

Dwg.10/13

Title Terms: COMPUTER; **MEMORY** ; STORAGE; METHOD; GENETIC; ALGORITHM; OPTIMUM; **MEMORY** ; LOCATE; INITIAL; POPULATION; SIZE; NUMBER; GENERATE; SELECT; INCREASE; POPULATION; SIZE; CONVERGE; ALGORITHM; NUMBER; GENERATE

Derwent Class: T01; U21
International Patent Class (Main): G06F-012/00; G06F-015/18; H03M-007/30
International Patent Class (Additional): G06F-015/00; G06F-017/00
File Segment: EPI

16/5/21 (Item 9 from file: 350)
DIALOG(R)File 350:Derwent WPIX

010594040 **Image available**
WPI Acc No: 1996-090993/ **199610**
XRPX Acc No: N97-327907
Computer based method for retrieving TRIE **dictionaries - constructing fixed-length backward** TRIE **dictionary for each of left sub-strings of each of number of character strings, which are constituents of forward** TRIE **dictionary**
Patent Assignee: IBM JAPAN LTD (IBMC); INT BUSINESS MACHINES CORP (IBMC)
Inventor: ITO N
Number of Countries: 002 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 7319900	A	19951208	JP 94108186	A	19940523	199610	B
US 5655129	A	19970805	US 95395731	A	19950228	199737	

Priority Applications (No Type Date): JP 94108186 A 19940523
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 7319900	A		10		
US 5655129	A		11		

Abstract (Basic): US 5655129 A
The method involves constructing a forward **TRIE** dictionary (Japanese forms of spelling) from a number of character strings and storing the resulting forward **TRIE** dictionary in a computer **memory** or media. A fixed-length backward **TRIE** dictionary is constructed for each of the left sub-strings of each of the number of character strings which are the constituents of the forward **TRIE** dictionary. The fixed-length backward **TRIE** dictionary begins with the last character of the left sub-string and ends with the first character of the left sub-string. A candidate character string lattice is then inputted.
E.g. when the candidate character string lattice comprises M columns (M - integer number), the work quantity for backward **TRIE** dictionary retrieval is calculated from a column k (k integer number), for each of k=1,,,M, for determining the column k in which the cost is minimum.
USE/ADVANTAGE - For retrieving Japanese language dictionary with **tree** structure called **TRIE** . Capable of high speed retrieval even when retrieving character string such as one with hawing wild card in prefix portion of input character string large number of candidate character.
Dwg.4/4
Title Terms: COMPUTER; BASED; METHOD; RETRIEVAL; DICTIONARY; CONSTRUCTION; FIX; LENGTH; BACKWARD; DICTIONARY; LEFT; SUB; STRING; NUMBER; CHARACTER; STRING; CONSTITUENT; FORWARD; DICTIONARY
Derwent Class: T01
International Patent Class (Main): G06F-003/14; G06F-017/30
File Segment: EPI

16/5/22 (Item 10 from file: 350)
DIALOG(R)File 350:Derwent WPIX

010580664 **Image available**
WPI Acc No: 1996-077617/ **199608**
XRPX Acc No: N96-064535

Variable length data sequence matching method for searching matching digital sequences in routing devices of communications networks - using trie -like database in which each node contains link or parent pointer to immediate predecessor node at next higher level of hierarchy which divides search process into two parts performed sequentially

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: DOERINGER W; DYKEMAN D; KARJOTH G; NASSEHI M; SHARMA M B; SHARMA M
Number of Countries: 018 Number of Patents: 005
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9600945	A1	19960111	WO 94EP2135	A	19940630	199608	B
EP 804769	A1	19971105	EP 94924215	A	19940630	199749	
			WO 94EP2135	A	19940630		
US 5787430	A	19980728	WO 94EP2135	A	19940630	199837	
			US 96765764	A	19961217		
EP 804769	B1	20000202	EP 94924215	A	19940630	200011	
			WO 94EP2135	A	19940630		
DE 69422935	E	20000309	DE 622935	A	19940630	200019	
			EP 94924215	A	19940630		
			WO 94EP2135	A	19940630		

Priority Applications (No Type Date): WO 94EP2135 A 19940630
Cited Patents: 01Jnl.Ref; EP 408188; EP 419889
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 9600945	A1	E	42	G06F-017/30	
Designated States (National): JP US					
Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE					
EP 804769	B1	E		G06F-017/30	Based on patent WO 9600945
Designated States (Regional): DE FR GB					
DE 69422935	E			G06F-017/30	Based on patent EP 804769
					Based on patent WO 9600945
EP 804769	A1	E		G06F-017/30	Based on patent WO 9600945
Designated States (Regional): DE FR GB					
US 5787430	A			G06F-017/30	Based on patent WO 9600945

Abstract (Basic): WO 9600945 A

The method of retrieving a partial match of a search argument (input key) from entries stored in a database having a **trie** -like structure with nodes (20) each containing link information (21) leading to at least one previous node (parent pointer), and second link information (25,26) leading to at least one following node (child pointer), at least one stored key (entry, 23,24) or a combination of the two, involves entering at a node of the database (root node). A search path is determined from one node to another through the **trie** -like database by successively processing segments of the search argument, and the second link information (25,26), until the segments are consumed or a (leaf) node lacking the second link information (25,26) is reached.

An entry stored in the node at which the search path ended is compared with the search argument, and if no partial match between the search argument and the entry is found in the current node. The method further involves back-tracking the search path by processing the first link information (21) of the current node. The method is repeated until at least a partial match is found or the root node is reached.

USE/ADVANTAGE - Retrieving partial matches of search argument from entries stored in database. Node structure allows two step search process which allows efficient use of **memories** , and enables fast data retrieval in communication within computer networks.

Dwg.2/4b

Title Terms: VARIABLE; LENGTH; DATA; SEQUENCE; MATCH; METHOD; SEARCH; MATCH ; DIGITAL; SEQUENCE; ROUTE; DEVICE; COMMUNICATE; NETWORK; DATABASE; NODE; CONTAIN; LINK; PARENT; POINT; IMMEDIATE; PREDECESSOR; NODE; HIGH; LEVEL; **HIERARCHY** ; DIVIDE; SEARCH; PROCESS; TWO; PART; PERFORMANCE; SEQUENCE
ent Class: T01; W01
rnational Patent Class (Main): G06F-017/30

File Segment: EPI

16/5/23 (Item 11 from file: 350)
DIALOG(R)File 350:Derwent WPIX

009857582 **Image available**
WPI Acc No: 1994-137438/ **199417**
XRPX Acc No: N94-107962

Multi-bit input address look-up method for packet data system - searching relatively large database of addresses using combination of programmable hash and binary search algorithms, and retrieving new stored address for match of stored and input hash addresses

Patent Assignee: CABLETRON SYSTEMS INC (CABL-N); DIGITAL EQUIP CORP (DIGI)
Inventor: SPINNEY B A
Number of Countries: 005 Number of Patents: 004
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 594196	A1	19940427	EP 93117155	A	19931022	199417	B
US 5414704	A	19950509	US 92964738	A	19921022	199524	
			US 94223379	A	19940405		
EP 594196	B1	19990331	EP 93117155	A	19931022	199917	
DE 69324204	E	19990506	DE 624204	A	19931022	199924	
			EP 93117155	A	19931022		

Priority Applications (No Type Date): US 92964738 A 19921022; US 94223379 A 19940405
Cited Patents: DE 4023527; EP 522743; US 4032987
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 594196	A1	E	29	H04L-012/46	
Designated States (Regional): DE FR GB IT					
US 5414704	A		23	H04J-003/26	Cont of application US 92964738
EP 594196	B1	E		H04L-012/46	
Designated States (Regional): DE FR GB IT					
DE 69324204	E			H04L-012/46	Based on patent EP 594196

Abstract (Basic): EP 594196 A

The appts. carries out source address and destination address look-ups for use in a packet data communication system. The appts. includes a controller (10) for interfacing between a fibre distributed data interface (FDDI) link (11) and a crossbar switch device (12). The crossbar switch has a number of input and output ports (13), each of which may be connected by another controller to another network segment (11) such as a FDDI link.

The controller contains a processor (20) for executing various processes which include accessing the packet **memory** (21) which stores incoming and outgoing packet data queues and translation and hash tables. The processor also accesses the content addressable **memory** (23) for use in exact matching of certain 48 bit source addresses.

ADVANTAGE - Inexpensive approach to very high speed address look-ups as is required in bridges and routers on high speed links in packet data communication networks. Requires on average two read operations.

Dwg.1A/8

Title Terms: MULTI; BIT; INPUT; ADDRESS; LOOK-UP; METHOD; PACKET; DATA; SYSTEM; SEARCH; RELATIVELY; DATABASE; ADDRESS; COMBINATION; PROGRAM; HASH ; BINARY; SEARCH; ALGORITHM; RETRIEVAL; NEW; STORAGE; ADDRESS; MATCH; STORAGE; INPUT; HASH; ADDRESS
Index Terms/Additional Words: FDDI
Derwent Class: T01; W01
International Patent Class (Main): H04J-003/26; H04L-012/46
International Patent Class (Additional): H04L-012/56
File Segment: EPI

16/5/24 (Item 12 from file: 350)
DIALOG(R)File 350:Derwent WPIX

009533352 **Image available**
WPI Acc No: 1993-226893/ **199328**
XRPX Acc No: N93-174208

Variable length decoder using combinational circuits or ROMs - decodes pruned binary trie **, and partitions decoding into segments, beginning with most significant bits, with each segment outputting valid code or informing next segment**

Patent Assignee: AMERICAN TELEPHONE & TELEGRAPH CO (AMTT); AT & T CORP (AMTT); AT & T BELL LAB (AMTT)
Inventor: KUSTKA G J
Number of Countries: 008 Number of Patents: 010
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5226082	A	19930706	US 92907977	A	19920702	199328	B
EP 577330	A2	19940105	EP 93304929	A	19930624	199402	
CA 2096176	A	19940103	CA 2096176	A	19930616	199412	
JP 6097838	A	19940408	JP 93183402	A	19930630	199419	
EP 577330	A3	19940601	EP 93304929	A	19930624	199525	
EP 577330	B1	19970903	EP 93304929	A	19930624	199740	
DE 69313540	E	19971009	DE 613540	A	19930624	199746	
			EP 93304929	A	19930624		
CA 2096176	C	19980908	CA 2096176	A	19930616	199846	
JP 3016996	B2	20000306	JP 93183402	A	19930630	200016	
KR 286195	B	20010416	KR 9312044	A	19930630	200218	

Priority Applications (No Type Date): US 92907977 A 19920702
Cited Patents: 3.Jnl.Ref; EP 426429; EP 467678; US 4816914
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5226082	A		13	H04L-009/00	
EP 577330	A2	E	15	H03M-007/40	
Designated States (Regional): DE FR GB IT					
CA 2096176	A			H03M-007/40	
JP 6097838	A		12	H03M-007/42	
EP 577330	A3			H04L-009/00	
EP 577330	B1	E	18	H03M-007/40	
Designated States (Regional): DE FR GB IT					
DE 69313540	E			H03M-007/40	Based on patent EP 577330
CA 2096176	C			H03M-007/40	
JP 3016996	B2		13	H03M-007/42	Previous Publ. patent JP 6097838
KR 286195	B			H03M-007/40	

Abstract (Basic): US 5226082 A

The appts. for decoding a stream of signals encoded with given a variable-length code (VLC) includes a first device which presents a number of bits of the stream of signals, which number is at least equal to the number of bits in the longest code of the VLC. A number of processing blocks are interconnected in a chain such that each block in the chain, other than the last one feeds information to a next block. Each block is responsive to a different group of adjacent bits presented by the first device and, in response to received information from the previous block and the group of adjacent bits, identifies a subset of the VLC.

Each processing block comprises a read-only **memory** responsiv to at least a subset of the group of adjacent bits, and decodes a portion of a **trie** that is pruned down to its k-nodes. A k-node is a node of the **trie** which supports a binary number of leaves of any particular number code length and which has no k-nodes in its path toward the root.

ADVANTAGE - Decoding complexity reduced with minimised number of k-nodes. May be implemented with pipelined architecture and using single **ROM** in processing block.

Dwg.5/7

Title Terms: VARIABLE; LENGTH; DECODE; COMBINATION; CIRCUIT; **ROM** ; DECODE;

PRUNE; BINARY; PARTITION; DECODE; SEGMENT; BEGIN; SIGNIFICANT; BIT; SEGMENT; OUTPUT; VALID; CODE; INFORMATION; SEGMENT
Derwent Class: W01; W04
International Patent Class (Main): H03M-007/40; H03M-007/42; H04L-009/00
File Segment: EPI

16/5/26 (Item 14 from file: 350)
DIALOG(R)File 350:Derwent WPIX

008270671 **Image available**
WPI Acc No: 1990-157672/ **199021**
XRPX Acc No: N90-122550
Bucket **-oriented route planning method - anticipating which** buckets **will be of importance in near future for calculation of navigation data**
Patent Assignee: PHILIPS GLOEILAMPENFAB NV (PHIG); US PHILIPS CORP (PHIG)
Inventor: VERSTRAETE R A; VERSTRAETE R
Number of Countries: 012 Number of Patents: 008
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 369539	A	19900523	EP 89202867	A	19891113	199021	B
NL 8802833	A	19900618				199028	
US 5170353	A	19921208	US 89366803	A	19890614	199252	
			US 91723704	A	19910625		
EP 369539	B1	19930519	EP 89202867	A	19891113	199320	
DE 68906648	E	19930624	DE 606648	A	19891113	199326	
			EP 89202867	A	19891113		
ES 2041401	T3	19931116	EP 89202867	A	19891113	199350	
JP 2996347	B2	19991227	JP 89296043	A	19891114	200006	
KR 159922	B1	19981215	KR 8916522	A	19891115	200034	

Priority Applications (No Type Date): NL 882833 A 19881117
Cited Patents: 2.Jnl.Ref
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 369539	A		15		
Designated States (Regional): CH DE ES FR GB IT LI SE					
US 5170353	A		10	G06F-015/50	Cont of application US 89366803
EP 369539	B1	E	13	G01C-021/20	
Designated States (Regional): CH DE ES FR GB IT LI SE					
DE 68906648	E			G01C-021/20	Based on patent EP 369539
ES 2041401	T3			G01C-021/20	Based on patent EP 369539
JP 2996347	B2		11	G08G-001/0969	Previous Publ. patent JP 2201600
KR 159922	B1			G06F-015/50	

Abstract (Basic): EP 369539 A
The optimum route-determing method uses topographical and traffic information and repeatally selects vectors and expands a search **tree** which contains vectors wwhich form already planned sub-routes. To each vector there is assigned a weighting factor and for each sub-route there is determined a cumulative weighting factor by addition of the weighing factors of the vectors of the already planned sub-route. The **information** is sub-divided into **buckets** , for the repeated selection of vectors and the expansion of the search **tree** there being used exclusively vectors from a predetermined maximum number of **buckets** which are selected from all **buckets** avaialbale on the basis of evaluation value.
The evaluation value is determined by a sum of the weighting factors of the constituent vectors of an already planned sub-route and an estimated fictitious sub-route yet to be followed via the relevant **bucket** .
USE - Cars.
Dwg.1/4
Title Terms: **BUCKET** ; ORIENT; ROUTE; PLAN; METHOD; ANTICIPATE; **BUCKET** ; IMPORTANT; FUTURE; CALCULATE; NAVIGATION; DATA
Derwent Class: P85; S02; T01; W06; X22

File 347:JAPIO Nov 1976-2004/May(Updated 040903)
(c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200459
(c) 2004 Thomson Derwent

Set	Items	Description
S1	1025151	MEMOR??? OR RAM OR DRAM OR SRAM OR SDRAM OR RDRAM OR SLDRAM OR SGRAM OR DRDRAM OR ROM OR PROM OR EPROM OR EEPROM OR FPO - OR EDO
S2	76619	DIRECTORY OR DIRECTORIES OR HIERARCH? OR TREE? ?
S3	2	TRIE()NODE? ?
S4	364520	TABLE? ? OR LUT? ?
S5	47	(POINT??? OR ADDRESS???)(5N)((LOWER OR DEEPER)(3N)NODE? ?)
S6	1137	BUCKET? ?(10N)(DATA OR INFORMATION OR POINT??? OR ADDRESS?- ?? OR S2)
S7	0	S1 AND S2 AND TRIE? ? AND BUCKET? ?
S8	23	S1 AND S2 AND TRIE? ?
S9	3	S8 AND S4:S5
S10	9	S1 AND S2 AND S6
S11	13	S1 AND S2 AND BUCKET? ?
S12	36	S8:S11
S13	12	S12 AND AC=US/PR
S14	7	S13 AND AY=(1970:1998)/PR
S15	25	S12 AND PY=1970:1998
S16	27	S14:S15

16/5/1 **(Item 1 from file: 347)**
DIALOG(R)File 347:JAPIO

05957492 **Image available**
DEVICE AND METHOD FOR SHOWING DEPENDENCY RELATION OF FILE

PUB. NO.: 10-240592 [JP 10240592 A]
PUBLISHED: September 11, 1998 (**19980911**)
INVENTOR(s): NEGISHI TAKASHI
MIYAGAWA REI
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 09-220303 [JP 97220303]
FILED: August 15, 1997 (19970815)
INTL CLASS: [6] G06F-012/00; G06F-003/14
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- **Memory** Units); 45.3 (INFORMATION PROCESSING -- Input Output Units

ABSTRACT

PROBLEM TO BE SOLVED: To provide a system which prevents a necessary file or folder from being deleted or moved carelessly by performing output indicating the dependency relation between files when the file or folder is operated.

SOLUTION: If a user **tries** to throw a file 23 in a trash can 24 or change the position where the file 23 is present while files 22 and 23, a **directory** (folder) 21, etc., are displayed in the form of icons, a thread display 25 is made between the icons to distinctively indicate that the file 22 used when the file 23 being operated is executed requires the file 23. Consequently, it is shown that they are relative files and an alarm sound is generated once the file is put in a specific area including the icon of the trash can 24. Further, the alarm sound is also generated when the user **tries** to move the file 22 to a place where the file 23 can not be found.

16/5/2 **(Item 2 from file: 347)**
DIALOG(R)File 347:JAPIO

05342142 **Image available**
SHARED **MEMORY** TYPE MULTIPROCESSOR SYSTEM

PUB. NO.: 08-297642 [JP 8297642 A]
PUBLISHED: November 12, 1996 (**19961112**)
INVENTOR(s): HORIKAWA KOICHI
APPLICANT(s): KOFU NIPPON DENKI KK [000000] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 07-102532 [JP 95102532]
FILED: April 26, 1995 (19950426)
INTL CLASS: [6] G06F-015/16; G06F-015/163
JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- Computer Applications)

ABSTRACT

PURPOSE: To improve performance by improving the hit rate of cache by guaranteeing coherency between caches concerning the **directory** system shared **memory** type multiprocessor system provided with plural store-in caches.

CONSTITUTION: **Directory** control parts 100 and 200 monitor requests on system buses 40 and 41 of their own clusters and transfer requests or data to the **directory** control part of the other cluster as needed. When the address competition discrimination circuit detects the coincidence of addresses between the request on the system bus of the present cluster and the request from the **directory** control part of the other cluster, 'cancel' is issued onto the system bus of the present cluster and the request issued onto the system bus of the present cluster is **tried** again. Besides, the system buses 40 and 41 are controlled while shifting their

cycles by half with each other.

16/5/3 (Item 3 from file: 347)
DIALOG(R)File 347:JAPIO

04559272 **Image available**
RETRIEVING SYSTEM

PUB. NO.: 06-231172 [JP 6231172 A]
PUBLISHED: August 19, 1994 (**19940819)**
INVENTOR(s): HIRANO YASUHIRO
MIURA FUMIMITSU
KOBAYASHI NOBUYUKI
APPLICANT(s): NIPPON TELEGR & TELEPH CORP <NTT> [000422] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 05-015153 [JP 9315153]
FILED: February 02, 1993 (19930202)
INTL CLASS: [5] G06F-015/40; G06F-013/00
JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- Computer Applications); 45.2 (INFORMATION PROCESSING -- **Memory** Units)
JOURNAL: Section: P, Section No. 1830, Vol. 18, No. 614, Pg. 131, November 22, 1994 (19941122)

ABSTRACT

PURPOSE: To shorten time required for inserting data and to improve simultaneous traveling efficiency between plural processes.

CONSTITUTION: An accessing means is formed by a hash function calculating means for calculating a hash value from a key value, **pointers** for storing the storing places of **data** or **buckets** 3 to 6 for storing **data** 3-2 to 6-2 and local depth parts 3-1 to 6-1, 2(sup n) (n is an integer large area depth) enteries 1-2 to 1-9 (i.e., the O-th entry to 2(sup (n-1))th entry), a **directory** 1 having large area depth 1-1 and capable of changing the number of entries, and a working variable 2. Since **bucket** retrieval is repeated until a **bucket** whose local depth is less than a working variable value is detected at the time of retrieving data, the number of entries to be updated at the time of dividing a **bucket** can be reduced and the repeating frequency of **bucket** retrieval can be reduced by executing the maintenance of the **directory** 1.

16/5/7 (Item 7 from file: 347)
DIALOG(R)File 347:JAPIO

02911950
INFORMATION PROTECTING METHOD

PUB. NO.: 01-209550 [JP 1209550 A]
PUBLISHED: August 23, 1989 (**19890823)**
INVENTOR(s): MARUYAMA TADASHI
APPLICANT(s): FUJI ELECTRIC CO LTD [000523] (A Japanese Company or Corporation), JP (Japan)
FUJI FACOM CORP [470926] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 63-034178 [JP 8834178]
FILED: February 17, 1988 (19880217)
INTL CLASS: [4] G06F-012/14
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- **Memory** Units)
JOURNAL: Section: P, Section No. 962, Vol. 13, No. 517, Pg. 66, November 20, 1989 (19891120)

ABSTRACT

PURPOSE: To realize the protection of information by securing such constitution where an operating system OS checks user identification names and carries out error processing in case such a file and a **directory** that

are not defined to the inhibited user names are opened.

CONSTITUTION: When a file opening request is received, a OS confirms whether the file and the **directory** that received the opening requests are included or not in an area controlled by the OS. When the presence of the file and the **directory** are confirmed, it is checked whether or not a user of the identification name to which the access to said file and **directory** are permitted **tries** to open the file. If the user has an inhibited identification name, the error processing is carried out to inform that no file is available. Thus it is difficult to know all information on both file and **directory** names and therefore the information can be protected.

16/5/8 (Item 8 from file: 347)
DIALOG(R)File 347:JAPIO

02701755 **Image available**
FALL-BACK OPERATING SYSTEM IN CASE OF FAULT OF **MEMORY**

PUB. NO.: 63-318655 [JP 63318655 A]
PUBLISHED: December 27, 1988 (**19881227)**
INVENTOR(s): HIROYAMA SHIGEHIRO
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 62-155976 [JP 87155976]
FILED: June 23, 1987 (19870623)
INTL CLASS: [4] G06F-012/16
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- **Memory** Units)
JOURNAL: Section: P, Section No. 859, Vol. 13, No. 162, Pg. 24, April 19, 1989 (19890419)

ABSTRACT

PURPOSE: To prevent a work from being suspended, by starting a fall-back operation as an non-resident DB/DS, in case a fault has been generated in a resident destination **memory** , and switching it automatically to a resident operation after the **memory** has been released.
CONSTITUTION: When a DB/DS open task is started by procedures 11'-15', an AIM **directory** 2 is retrieved. As a result, for instance, it becomes clear that a resident DB/DS defined by a schema 1 is resident, therefore, from in a DB use DASD 3, a data base defined by the schema concerned 1 is opened. Subsequently, after an AIM resident DB/DS management **table** 4 has been generated, it is **tried** to execute loading to a resident DB/DS use area 1-1. When a **memory** fault is generated in this area, first of all, the AIM resident DB/DS concerned is opened as non-resident, and a fall-back operation is started. After the **memory** fault has been released, it is switched to a resident operation by loading to a resident destination, and it is prevented to suspend a work.

16/5/10 (Item 10 from file: 347)
DIALOG(R)File 347:JAPIO

02584832 **Image available**
SHARED RESOURCE MANAGEMENT PROCESSING SYSTEM

PUB. NO.: 63-201732 [JP 63201732 A]
PUBLISHED: August 19, 1988 (**19880819)**
INVENTOR(s): KITADATE YOTARO
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 62-033939 [JP 8733939]
FILED: February 17, 1987 (19870217)
INTL CLASS: [4] G06F-009/46; G06F-015/16
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units); 45.4 (INFORMATION PROCESSING -- Computer Applications)
JOURNAL: Section: P, Section No. 803, Vol. 12, No. 488, Pg. 167,

December 20, 1988 (19881220)

ABSTRACT

PURPOSE: To make a range in which an exclusive occupancy is generated as small as possible and also to make a main storage capacity comparatively small, by dividing and subdividing a common resources area and systematizing it to a **tree** shape, at the time of carrying forward a processing, while occupying the shared resources in dependently to each other.

CONSTITUTION: As for nodes A-E of a range which contains a root A in a **tree** struc ture, for instance, shown as a **tree** part .alpha. in the figure, the possibility that they are used by following up a fact that they are occupied themselves or one of the subordinate nodes is occupied is high, therefore, they are managed by a fixed management **table** 5. That is, management information is always held in a form of 1-to-1 in accordance with separate node. On the other hand, as for nodes F-K of a range shown as a **tree** part .beta. in the figure, the frequency by which they are occupied is not so high, and when it is **tried** to hold the management information in a form of 1-to-1 in accor dance with a separate node, sometimes the capacity of a main storage 1 become large. Therefore, when a state that the nodes are occupied as the present problem is generated, they are registered temporarily on a management **table** 6 and managed. In such a way, the common resources can be managed effectively, while reducing the **memory** capacity.

16/5/11 (Item 11 from file: 347)

01909851 **Image available**

MEMORY SYSTEM

PUB. NO.: 61-123951 [JP 61123951 A]
PUBLISHED: June 11, 1986 (**19860611)**
INVENTOR(s): AKIMOTO HARUO
SHIMIZU SHINICHI
SHINAGAWA AKIO
SATO KIMINORI
YASUSATO AKIRA
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 59-245528 [JP 84245528]
FILED: November 20, 1984 (19841120)
INTL CLASS: [4] G06F-012/04; G06F-012/06
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- **Memory** Units)
JOURNAL: Section: P, Section No. 509, Vol. 10, No. 312, Pg. 130, October 23, 1986 (19861023)

ABSTRACT

PURPOSE: To speed up the preparation to start the processing, by using **memory** elements having different access time as a **memory** to store the bit row which consists of one word, and by making the access speed **hierarchical** by each access unit.

CONSTITUTION: The first 8 bits + 1 parity of a word area stored in each address of a high speed **memory** 9, and the remaining part is stored at each address of low speed **memory** 10. For example, when a central processing unit 11 **tries** an access to a variable A, it specifies the addresses o the variable A and issue the read requests to the high speed **memories** 9 and the low speed **memory** 10 at the same time. Then, the tag information is first sent from the high speed **memory** 9, and the processing device 11 checks the data type, and prepares for the next processing according to the type. About 1-2 clocks later, a data is sent from the low speed **memory** 10, and at this moment, the preparation has been already completed, then it can immediately proceed to the next processing.

PATENT COOPERATION TREATY

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference P00181PCT	FOR FURTHER ACTION	see Notification of Transmittal of International Search Report (Form PCT/ISA/220) as well as, where applicable, item 5 below.
International application No. PCT/FI 99/00717	International filing date *(day/month/year)* 2 Sept 1999	(Earliest) Priority Date *(day/month/year)* 29 Sept 1998
Applicant Nokia Networks OY et al		

This international search report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This international search report consists of a total of 3 sheets.

[X] It is also accompanied by a copy of each prior art document cited in this report.

1. [ ] Certain claims were found unsearchable (See Box I).

2. [ ] Unity of invention is lacking (See Box II).

3. [ ] The international application contains disclosure of a nucleotide and/or amino acid sequence listing and the international search was carried out on the basis of the sequence listing
   - [ ] filed with the international application.
   - [ ] furnished by the applicant separately from the international application,
     - [ ] but not accompanied by a statement to the effect that it did not include matter going beyond the disclosure in the international application as filed.
   - [ ] transcribed by this Authority.

4. With regard to the title,
   - [ ] the text is approved as submitted by the applicant.
   - [X] the text has been established by this Authority to read as follows:

   Compression of nodes in a trie structure

5. With regard to the abstract,
   - [X] the text is approved as submitted by the applicant.
   - [ ] the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box III. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. The figure of the drawings to be published with the abstract is:
   Figure No. 3
   - [X] as suggested by the applicant.
   - [ ] because the applicant failed to suggest a figure.
   - [ ] because this figure better characterizes the invention.
   - [ ] None of the figures.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI 99/00717

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: G06F 17/30
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 9841932 A1 (NOKIA TELECOMMUNICATIONS OY), 24 Sept 1998 (24.09.98), page 1, line 1 - page 4, line 18, claims 1-14, abstract	1,8,12,19, 23-26
A	page 1, line 1 - page 4, line 18, claims 1-14, abstract	2-7,9-11, 13-18,20-22
	--	
Y	US 5276868 A (NIGEL T. POOLE), 4 January 1994 (04.01.94), column 1, line 1 - column 3, line 30, figure 2, abstract	1,8,12,19, 23-26
A	column 1, line 1 - column 3, line 30, figure 2, abstract	2-7,9-11, 13-18,20-22
	--	

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" erlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 March 2000	17 -03- 2000
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Oskar Pihlgren/CL Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI 99/00717

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5721899 A (ISAO NAMBA), 24 February 1998 (24.02.98), see whole document -- --------	1-26

INTERNATIONAL SEARCH REPORT

Information on patent family members

02/12/99

International application No.
PCT/FI 99/00717

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9841932 A1	24/09/98	AU 6623998 A	12/10/98
		FI 102425 B	00/00/00
		FI 971066 A	15/09/98
US 5276868 A	04/01/94	CA 2043028 A	24/11/91
		EP 0458698 A	27/11/91
		JP 6004585 A	14/01/94
US 5721899 A	24/02/98	JP 8194719 A	30/07/96

```
File   8:Ei Compendex(R) 1970-2004/Sep W2
         (c) 2004 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2004/Aug
         (c) 2004 ProQuest Info&Learning
File 202:Info. Sci. & Tech. Abs. 1966-2004/Sep 09
         (c) 2004 EBSCO Publishing
File  65:Inside Conferences 1993-2004/Sep W3
         (c) 2004 BLDSC all rts. reserv.
File   2:INSPEC 1969-2004/Sep W2
         (c) 2004 Institution of Electrical Engineers
File  94:JICST-EPlus 1985-2004/Aug W4
         (c)2004 Japan Science and Tech Corp(JST)
File 483:Newspaper Abs Daily 1986-2004/Sep 17
         (c) 2004 ProQuest Info&Learning
File   6:NTIS 1964-2004/Sep W2
         (c) 2004 NTIS, Intl Cpyrght All Rights Res
File 144:Pascal 1973-2004/Sep W2
         (c) 2004 INIST/CNRS
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
File  34:SciSearch(R) Cited Ref Sci 1990-2004/Sep W2
         (c) 2004 Inst for Sci Info
File  99:Wilson Appl. Sci & Tech Abs 1983-2004/Aug
         (c) 2004 The HW Wilson Co.
File 583:Gale Group Globalbase(TM) 1986-2002/Dec 13
         (c) 2002 The Gale Group
File  95:TEME-Technology & Management 1989-2004/Jun W1
         (c) 2004 FIZ TECHNIK
File 438:Library Lit. & Info. Science 1984-2004/Aug
         (c) 2004 The HW Wilson Co

Set     Items   Description
S1     884385   MEMOR??? OR RAM OR DRAM OR SRAM OR SDRAM OR RDRAM OR SLDRAM
              OR SGRAM OR DRDRAM OR ROM OR PROM OR EPROM OR EEPROM OR FPO -
             OR EDO
S2     805692   DIRECTORY OR DIRECTORIES OR HIERARCH? OR TREE? ?
S3          6   TRIE()NODE? ?
S4     614613   TABLE? ? OR LUT? ?
S5          7   (POINT??? OR ADDRESS???)(5N)((LOWER OR DEEPER)(3N)NODE? ?)
S6       1279   BUCKET? ?(10N)(DATA OR INFORMATION OR POINT??? OR ADDRESS?-
              ?? OR S2)
S7          0   S1 AND S2 AND S3 AND S6
S8          0   S1 AND S2 AND S3 AND BUCKET? ?
S9         81   S1 AND S2 AND (S3 OR BUCKET? ?)
S10         0   S1 AND S2 AND S3
S11       308   S1 AND S2 AND TRIE? ?
S12        12   S9 AND S11
S13         2   S1 AND S3
S14       229   COMPRESS?(3N)NODE? ?
S15         3   S14 AND (TRIE? ? OR BUCKET? ?)
S16        37   S11 AND S4
S17         0   S11 AND S5
S18        38   BUCKET? ?(3N)NODE? ?
S19        21   S1:S2 AND S18
S20        88   S12:S13 OR S15:S19
S21        54   RD (unique items)
S22        29   S21 NOT PY=1999:2004
S23       500   AU=(IIVONEN J? OR IIVONEN, J? OR TIKKANEN, M? OR TIKKANEN -
             M?)
S24         7   S23 AND (TRIE? ? OR BUCKET? ?)
S25         2   RD (unique items)
```

22/5/1 (Item 1 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

04644251 E.I. No: EIP97033565290
Title: Average height of a node in the BANG abstract directory tree
Author: Taylor, Stephen; Hachem, Nabil; Selkow, Stanley
Corporate Source: Worcester Polytechnic Inst, Worcester, MA, USA
Source: Information Processing Letters v 61 n 1 Jan 14 1997. p 55-61
Publication Year: 1997
CODEN: IFPLAT ISSN: 0020-0190
Language: English
Document Type: JA; (Journal Article) Treatment: G; (General Review); T; (Theoretical)
Journal Announcement: 9704W5
Abstract: The abstract logical data structure for the BANG file **directory** is a multiway **tree** structure with one **node** for each **bucket** in the file. Under assumptions of `perfect hashing' or `growth on data principle', we model the growth of the **tree** . The average cost for search and insertion is found to be logarithmic in the file size. The order constant is small and depends on the capacity of a bucket. Simulation confirms the analytic results. Similar assumptions should be applicable to the analysis of other multi-dimensional file structures. (Author abstract) 13 Refs.
Descriptors: Data structures; **Trees** (mathematics); Algorithms; Database systems; Computer simulation
Identifiers: Multiway **tree** structure; Multi dimensional file structure; BANG file **directory**
Classification Codes:
723.2 (Data Processing); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 723.3 (Database Systems); 723.5 (Computer Applications)
723 (Computer Software); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

22/5/2 (Item 2 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

04312252 E.I. No: EIP95122967565
Title: Novel congestion control in hybrid ATM/TDMA networks
Author: Talla, Malleswara; Elhakeem, Ahmed K.; Kadoch, Michel
Corporate Source: SITA, Montreal, Que, Can
Source: Computers & Electrical Engineering v 21 n 6 Nov 1995. p 397-416
Publication Year: 1995
CODEN: CPEEBQ ISSN: 0045-7906
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)
Journal Announcement: 9602W3
Abstract: A new congestion control scheme is analyzed for an ATM multiplexer node. This scheme is based on the leaky bucket and virtual leaky bucket techniques, and utilizes the interaction between the ATM and higher layers, in a hybrid asynchronous transfer mode/time division multiple access (ATM/TDMA) network. The transport users are assumed to be generic ATM sources, who modulate their end-to-end flow control parameters, i.e. protocol data unit size in case of video and voice users, and window size in case of data users, based on the congestion status. Simple analytical formulas are derived for congestion criteria, to represent the required bandwidth to support various classes of service, i.e. video, voice, data, etc. with their own performance requirements. An ATM multiplexer node buffer is analyzed using a modulated poisson process queuing model with bulk arrival and bulk service of cells. The ATM multiplexer node congestion performance criteria, i.e. the mean probabilities of ATM multiplexer node congestion, cell generation, cell discarding, buffer content and buffer overflow, are evaluated with and without the congestion control schemes. (Author abstract) 9 Refs.

Descriptors: *Congestion control (communication); Asynchronous transfer mode; Broadband networks; Network protocols; Bandwidth; Mathematical models ; Queueing theory; Telecommunication services
Identifiers: Asynchronous transfer mode multiplexer **node** ; Admission control; Leaky **bucket** technique; Virtual leaky bucket technique; Protocol data unit
Classification Codes:
922.1 (Probability Theory)
716 (Radar, Radio & TV Electronic Equipment); 717 (Electro-Optical Communications); 718 (Telephone & Line Communications); 723 (Computer Software); 921 (Applied Mathematics); 922 (Statistical Methods)
71 (ELECTRONICS & COMMUNICATIONS); 72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

22/5/3 (Item 3 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

04056673 E.I. No: EIP95022546650
Title: Join workload partitioning under uniform and skewed input relations
Author: Barlos, Fotios; Frieder, Ophir
Corporate Source: Thinking Machines Corp, Cambridge, MA, USA
Source: Parallel Processing Letters 4 1-2 June 1994. p 95-104
Publication Year: 1994
CODEN: PPLTEE
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 9504W2
Abstract: Parallel Join algorithms partition the workload into buckets and assign each **bucket** to a **node** of the multiprocessor system. The existing algorithms use the volume of the load as a metric to determine the bucket boundaries. When the input relations exhibit a high degree of skew, the above metric does not achieve uniform partitioning. We propose a new method to partition the workload of the Join operation that guarantees near equal execution time of the created buckets. We present results obtained from the Intel i860 hypercube system that support our theory. (Author abstract) 14 Refs.
Descriptors: *Multiprocessing systems; Algorithms; Distributed database systems; Computational methods; Computational complexity; Data processing
Identifiers: Join workload partitioning time; Parallel databases; Skewed data; Bucket boundaries
Classification Codes:
722.4 (Digital Computers & Systems); 723.3 (Database Systems); 721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory); 723.2 (Data Processing)
722 (Computer Hardware); 921 (Applied Mathematics); 723 (Computer Software); 721 (Computer Circuits & Logic Elements)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

22/5/4 (Item 4 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

04025895 E.I. No: EIP94122502674
Title: Time- memory **tradeoffs in vector quantizer codebook searching based on decision** trees
Author: Moayeri, Nader; Neuhoff, David L.
Corporate Source: Rutgers Univ, Piscataway, NJ, USA
Source: IEEE Transactions on Speech and Audio Processing v 2 n 4 Oct 1994. p 490-506
Publication Year: 1994
CODEN: IESPEJ ISSN: 1063-6676
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)

Journal Announcement: 9503W1

Abstract: This paper presents several algorithms for designing fixed- and variable-depth decision **trees** for searching vector quantizer (VQ) codebooks. Two applications of such are explored. First, given a source vector, a **tree** can be used to find the closest codevector in the VQ codebook with many fewer arithmetic operations than the usual 'full search.' This decrease in complexity comes at the expense of an increase in auxiliary table storage. Second, the **tree** can be used as the first stage of fine-coarse vector quantization, which yields further savings in complexity at the cost of somewhat more storage and a small increase in distortion. The design methods involve incrementally growing **trees** with a variety of node splitting criteria and, subsequently, optimally pruning **trees** on the basis of performance functionals such as distortion, storage, and computational complexity. The pruning is done with the BFOS algorithm, which optimally trades one performance functional with another, and with an extension of the BFOS algorithm wherein one performance measure is traded with a combination of two others. The results of applying these methods to i.i.d. Gaussian, Gauss-Markov, and sampled speech sources at encoding rates of one and two bits per source sample demonstrate the tradeoffs achievable amongst time (complexity), **memory** (storage), and distortion. (Author abstract) 23 Refs.

Descriptors: Algorithms; Decision theory; **Trees** (mathematics); Vectors; Computational complexity; Data storage equipment; Performance; Terminology; Encoding (symbols); Signal distortion

Identifiers: Vector quantizer codebook; Decision **trees** ; Time **memory** tradeoffs; **Node** splitting criteria; **Bucket**

Classification Codes:

921.6 (Numerical Methods); 721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 921.1 (Algebra); 722.1 (Data Storage, Equipment & Techniques); 723.2 (Data Processing)

921 (Applied Mathematics); 721 (Computer Circuits & Logic Elements); 722 (Computer Hardware); 723 (Computer Software)

92 (ENGINEERING MATHEMATICS); 72 (COMPUTERS & DATA PROCESSING)

22/5/5 (Item 5 from file: 8)

03798466 E.I. No: EIP94021199473

Title: Using difficulty of prediction to decrease computation: Fast sort, priority queue and convex hull on entropy bounded inputs

Author: Chen, Shenfeng; Reif, John H.

Corporate Source: Duke Univ, Durham, NC, USA

Conference Title: Proceedings of the 34th Annual Symposium on Foundations of Computer Science

Conference Location: Palo Alto, CA, USA Conference Date: 19931103-19931105

Sponsor: IEEE Computer Society

E.I. Conference No.: 19617

Source: Annual Symposium on Foundations of Computer Science (Proceedings) 1993. Publ by IEEE, Computer Society Press, Los Alamitos, CA, USA,(IEEE cat n 93CH3368-8). p 104-112

Publication Year: 1993

CODEN: 001190 ISSN: 0272-5428 ISBN: 0-8186-4370-6

Language: English

Document Type: CA; (Conference Article) Treatment: A; (Applications); T ; (Theoretical)

Journal Announcement: 9403W4

Abstract: There is an upsurge in interest in the Markov model and also more general stationary ergodic stochastic distributions in theoretical computer science community recently (e.g. see left bracket Vitter, Krishnan91 right bracket , left bracket Karlin, Philips, Raghavan92 right bracket , left bracket Raghavan92 right bracket for use of Markov models for on-line algorithms, e.g., cashing and prefetching). Their results used the fact that compressible sources are predictable (and vise versa), and

showed that on-line algorithms can improve their performance by prediction. Actual page access sequences are in fact somewhat compressible, so their predictive methods can be of benefit. This paper investigates the interesting idea of decreasing computation by using learning in the opposite way, namely to determine the difficulty of prediction. That is, we will approximately learn the input distribution, and then improve the performance of the computation when the input is not too predictable, rather than the reverse. To our knowledge, this is first case of a computational problem where we do not assume any particular fixed input distribution and yet computation is decreased when the input is less predictable, rather than the reverse. We concentrate our investigation on a basic computational problem: sorting and a basic data structure problem: maintaining a priority queue. We present the first known case of sorting and priority queue algorithms whose complexity depends on the binary entropy H less than equivalent to 1 of input keys where assume that input keys are generated from an unknown but arbitrary stationary ergodic source. This is, we assume that each of the input keys can be each arbitrarily long, but have entropy H. Note that H can be estimated in practice since the compression ratio rho using optimal Ziv-Lempel compression limits to 1/H for large inputs. Although sets of keys found in practice can not be expected to satisfy any fixed particular distribution such as uniform distribution, there is a large well documented body of empirical evidence that shows this compression ratio rho and thus 1/H is a constant for realistic inputs encountered in practice, say typically around 3 to at most 20. Our algorithm runs in O(n log (log n/H)) sequential expected time to sort n keys in a unit cost sequential **RAM** machine. This is O(n log log n) with the very reasonable assumption that the compression ratio rho equals 1/H of the input keys is no more than log**O**(**1**)n. Previous sorting algorithms are all Omega (n log n) except those that (i) assume a bound on the length of each key or (ii) assume a fixed (e.g., uniform) distribution. Instead, we learn an approximation to an unknown probability distribution (which can be any stationary ergodic source, not necessarily a Markov source) of the input keys by randomized subsampling and then implicitly build a suffix **tree** using fast **trie** and hash **table** data structures. We can also apply this method for priority queue. Given a subsampling of size n/(log n)**O**(**1**) which we use to learn the distribution, we then have O(log(log n/H)) expected sequential time per priority queue operation, with no assumption on the length of a key. Also we show our sequential sorting algorithm can be optimally speed up by parallelization without increase in total work bounds (though the parallel time bounds depend on an assumed maximum length L of each key). In particular, if L less than equivalent to n**O**(**1**), we get O(log n) expected time using O(n log(log n/H)/log n) processors for parallel sorting of n keys on a CRCW PRAM. We also give an application of our sorting algorithm to 2-D convex hull problem proving the same parallel complexity bound for this problem as for sorting. We have implemented the sequential version of our sorting algorithm on SPARC-2 machine and compared to the UNIX system sorting routine-quick sort. We found that our algorithm beats quicksort for large n on extrapolated empirical data. Our algorithm is even more advantageous in applications where the keys are many words long. (Author abstract) 41 Refs.

Descriptors: Algorithms; Learning systems; Data structures; Sorting; Queueing theory; Data compression; Computational complexity; **Trees** (mathematics); Probability; Random processes

Identifiers: Markov model; Prediction difficulty; Priority queue; Binary entropy; Ziv Lempel compression; Suffix **tree** ; Convex hull problems

Classification Codes:

721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory); 723.1 (Computer Programming); 723.2 (Data Processing); 922.1 (Probability Theory); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 722.4 (Digital Computers & Systems)

721 (Computer Circuits & Logic Elements); 723 (Computer Software); 922 (Statistical Methods); 921 (Applied Mathematics); 722 (Computer Hardware)

72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

22/5/6 (Item 6 from file: 8)

DIALOG(R)File 8:Ei Compendex(R)

03790658 E.I. No: EIP94011195903
Title: On the development of a site selection optimizer for distributed and parallel database systems
Author: Barlos, Fotis; Frieder, Ophir
Corporate Source: Thinking Machines Corp, Houston, TX, USA
Conference Title: Proceedings of the 2nd International Conference on Information and Knowledge Management
Conference Location: Washington, DC, USA Conference Date: 19931101-19931105
Sponsor: ACM, SIGART; ACM, SIGIR; International Society of Computers and Applications
E.I. Conference No.: 19822
Source: Proc 2 Int Conf Inf Knowl Manage 1993. Publ by ACM, New York, NY, USA. p 684-693
Publication Year: 1993
ISBN: 0-89791-626-3
Language: English
Document Type: CA; (Conference Article) Treatment: A; (Applications); G ; (General Review)
Journal Announcement: 9403W2
Abstract: The continuous increase in the volume of data, decrees the employment of Parallel and Distributed Computing. The processing requirements of parallel environments are complex and more stringent than the uniprocessor systems. Efficient exploitation of parallelism dictates an even partitioning of the computation across the processing sites. To achieve a uniform load, the database optimizers require statistical information of the underlining relations. We developed a query optimization approach, named Dynamic Optimization on Multiprocessor Engines (DOME), that uses a dynamic sampling methodology to determine the frequency distribution along each level of the query **tree** . DOME covers the three main multiprocessor query optimization areas of Workload Partitioning, Site Selection, and Operation Ordering. We present the Site Selection segment of DOME. The Site Selection segment receives the statistical characteristics of the participating relations from the Workload Partitioning segment and performs three major operations: identifies an allocation scheme between **buckets** and **nodes** ; partitions the relations residing on the various processor of the distributed environment into the appropriate buckets; transfers the **buckets** to their corresponding **nodes** . We implemented DOME on an Intel i860 hypercube with 32 nodes and tested its behavior through experimentation. The Site Selection algorithms provide approximately a six fold factor improvement over a static allocation approach for the Join relational operation. (Author abstract) 20 Refs.
Descriptors: *Database systems; Storage allocation (computer); Data handling; Computational methods; Parallel processing systems; Optimization; Algorithms; Relational database systems; Distributed database systems; Data processing
Identifiers: Site selection optimizer; Parallel database systems; Dynamic optimization on multiprocessor engines; Data skew problem
Classification Codes:
723.3 (Database Systems); 723.2 (Data Processing); 722.1 (Data Storage, Equipment & Techniques); 921.6 (Numerical Methods); 722.4 (Digital Computers & Systems); 921.5 (Optimization Techniques)
723 (Computer Software); 722 (Computer Hardware); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

22/5/7 (Item 7 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

02881088 E.I. Monthly No: EI9004039399
Title: Concurrency and Trie **Hashing.**
Author: Litwin, W.; Sagiv, Y.; Vidyasankar, K.
Corporate Source: I.N.R.I.A., Le Chesney, Fr

Source: Acta Informatica v 26 n 7 1989 p 597-614
Publication Year: 1989
CODEN: AINFA2 ISSN: 0001-5903
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)
Journal Announcement: 9004
Abstract: The **Trie** Hashing (TH), defined by Litwin, is one of the fastest access methods for dynamic and ordered files. The hashing function is defined in terms of a **trie** , which is basically a binary **tree** where a character string is associated implicitly with each node. This string is compared with a prefix of the given key in the search process, and depending on the result either the left or the right child is chosen as the next node to visit. The leaf **nodes** point to **buckets** which contain the records. The **buckets** are on a disk, whereas the **trie** itself is in the core **memory** . In this paper we consider concurrent execution of the TH operations. In addition to the usual search, insertion and deletion operations, we also include range queries among the concurrent operations. (Edited author abstract) 12 Refs.
Descriptors: *COMPUTER SYSTEMS PROGRAMMING--*Multiprocessing Programs; COMPUTER PROGRAMMING--Algorithms
Identifiers: **TRIE** HASHING (TH)
Classification Codes:
723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING)

22/5/8 (Item 8 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

02798227 E.I. Monthly No: EI8910105906
Title: Computational study of efficient shortest path algorithms.
Author: Hung, Ming S.; Divoky, James J.
Corporate Source: Kent State Univ, Kent, OH, USA
Source: Computers & Operations Research v 15 n 6 1988 p 567-576
Publication Year: 1988
CODEN: CMORAP ISSN: 0305-0548
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications); X; (Experimental)
Journal Announcement: 8910
Abstract: Five efficient shortest path algorithms are implemented and compared in this report. The selected algorithms are the most efficient, measured either in terms of worst case bounds or from previous computational studies. The algorithms include two using threshold functions, two using heaps, and one using **buckets** for sorting **node** labels. The last three algorithms have not been studied in detail before. The computational experiment employs a rigorous design to ensure that the results have statistical validity. Three different cost functions are generated to measure the sensitivity of each algorithm to cost distributions. Curve fittings are performed to summarize the results and they show high degrees of goodness-of-fit. The results reveal some heretofore unknown properties of some of the algorithms. (Edited author abstract) 21 Refs.
Descriptors: OPERATIONS RESEARCH; STATISTICAL METHODS--Regression Analysis; COMPUTER AIDED ANALYSIS; MATHEMATICAL TECHNIQUES-- **Trees**
Identifiers: SHORTEST PATH ALGORITHMS; NETWORK FLOW PROBLEMS; COST FUNCTIONS; THRESHOLD FUNCTIONS
Classification Codes:
912 (Industrial Engineering & Management); 922 (Statistical Methods); 723 (Computer Software); 921 (Applied Mathematics)
91 (ENGINEERING MANAGEMENT); 92 (ENGINEERING MATHEMATICS); 72 (COMPUTERS & DATA PROCESSING)

22/5/9 (Item 9 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

02716191 E.I. Monthly No: EI8903020881
Title: Simple bounded disorder file organization with good performance.
Author: Lomet, David B.
Corporate Source: Wang Inst of Graduate Studies
Source: ACM Transactions on Database Systems v 13 n 4 Dec 1988 p 525-551
Publication Year: 1988
CODEN: ATDSD3 ISSN: 0362-5915
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 8903
Abstract: A bounded-disorder (BD) file is one in which data are organized into nodes that are indexed, e.g., by means of a B- **tree** . The data nodes are multibucket nodes that are accessed by hashing. In this paper we present two important improvements to the BD organization as originally described. First, records in a data node that overflow their designated primary bucket are stored in a single overflow bucket which is itself a **bucket** of the data **node** . Second, when file space needs to be increased, partial expansions are used that employ elastic buckets. Analysis and simulation results demonstrate that the variant of the BD organization has utilization, random access performance, and file growth performance that can be competitive with good extendible hashing methods, while supporting high-performance sequential access. The simplicity of the organization results in simple algorithms for realizing the organization. (Author abstract) 12 Refs.
Descriptors: *DATA PROCESSING--*File Organization; COMPUTER SIMULATION; COMPUTER PROGRAMMING--Algorithms
Identifiers: BOUNDED DISORDER FILE ORGANIZATION; DYNAMIC FILES; INDEX SEQUENTIAL ACCESS; INDEXED FILES; STORAGE MANAGEMENT
Classification Codes:
723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING)

22/5/10 (Item 10 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

02335516 E.I. Monthly No: EI8711112520
Title: SOME CHARACTERISTIC CURVES FOR DICTIONARY ORGANIZATION WITH DIGITAL SEARCH.
Author: Sinha, R. M. K.
Corporate Source: Inst Natl de la Recherche Scientifique, Verdun, Que, Can
Source: IEEE Transactions on Systems, Man and Cybernetics v SMC-17 n 3 May/Jun 1987 p 520-527
Publication Year: 1987
CODEN: ISYMAW ISSN: 0018-9472
Language: ENGLISH
Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)
Journal Announcement: 8711
Abstract: Some parameters that characterize the natural language text behavior for typical structuring of a dictionary have been defined. The typical structure of the dictionary considered here is based on trie structuring using digital search. Digital search is well-suited for applications like optical character recognition. The dictionary uses three partitions in its structure. The first partition carries most frequently used words completely represented in trie structure in the main **memory** . In the second partition only a part of word is stored in trie structure and the rest is stored in suitable tail structures also in the main **memory** . In the third partition, a still smaller part of the word is held in trie structure, and the rest of the words are held in files on secondary storage device. The parameters defined are the trie nonutility factor, giving a measure of effectiveness of trie structure; the streaming factor, giving a measure of the common part that exists at the beginning of the word; the node-utilization factor, giving a measure of the extent to which multilink

node structure is suited as **trie node** ; and the dispersion factor, giving a measure of the average number of elements in the tail structures. 7 refs.
Descriptors: *INFORMATION SCIENCE; INFORMATION RETRIEVAL SYSTEMS; CHARACTER RECOGNITION, OPTICAL; DATABASE SYSTEMS; DATA PROCESSING--Word Processing
Identifiers: SEARCH METHODS; TEXT PROCESSING; DICTIONARY ORGANIZATION; NATURAL LANGUAGE DICTIONARY; TRIE STRUCTURING
Classification Codes:
903 (Information Science); 723 (Computer Software)
90 (GENERAL ENGINEERING); 72 (COMPUTERS & DATA PROCESSING)

22/5/11 (Item 11 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

01561623 E.I. Monthly No: EI8409087989 E.I. Yearly No: EI84023177
Title: ON THE PERFORMANCE EVALUATION OF EXTENDIBLE HASHING AND TREE **SEARCHING.**
Author: Flajolet, Philippe
Corporate Source: Inst Natl de Recherche en Informatique et en Automatique, Le Chesnay, Fr
Source: Acta Informatica v 20 n 4 1983 p 345-369
Publication Year: 1983
CODEN: AINFA2 ISSN: 0001-5903
Language: ENGLISH
Journal Announcement: 8409
Abstract: A class of **trees** occurs both in digital searching and in schemes for maintaining dynamic has **tables** . The author studies the distribution of height in these **trees** using the saddle point method of complex analysis. As a result, a precise evaluation is derived of the **memory** requirements of extendible hashing - a dynamic hashing scheme - and some related implementation issues is discussed. 15 refs.
Descriptors: *COMPUTER PROGRAMMING
Identifiers: EXTENDIBLE HASHING; **TRIE** SEARCHING
Classification Codes:
723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING)

22/5/12 (Item 12 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

00846477 E.I. Monthly No: EI7909068718 E.I. Yearly No: EI79022397
Title: FIXED-BUCKET BINARY STORAGE TREES .
Author: Knott, Gary D.
Corporate Source: Natl Inst of Health, Bethesda, Md
Source: Proc Hawaii Int Conf Syst Sci 12th, Honolulu, Hawaii, Jan 4-5 1979. Publ by West Period Co, North Hollywood, Calif, 1979 v 1 p 36-48
Publication Year: 1979
CODEN: PHISD7 ISSN: 0073-1129
Language: ENGLISH
Journal Announcement: 7909
Abstract: A binary storage **tree** has a set or bucket of possible items associated with each **node** . The **buckets** at deeper levels are refinements of the partitionings at earlier levels. When these buckets are established a priori, rather than determined by the particular items stored, the author obtains a storage data structure which is a generalized binary digital **tree** as well as a binary storage **tree** . Thus the binary key-values of the items along a path in a fixed-bucket binary storage **tree** have successively longer common prefixes. This synthesis of two schemes inherits all the desirable properties of both methods. The method is analyzed for uniformly-distributed input and shown to have the same cost statistics as binary digital **trees** . 10 refs.
Descriptors: *DATA PROCESSING--*Data Structures
Classification Codes:

723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING)

22/5/13 (Item 1 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01470673 ORDER NO: AADAA-I9607773
SPATIAL DATA STRUCTURES AND QUERY PERFORMANCE IN THE SEQUENTIAL AND DATA-PARALLEL DOMAINS
Author: HOEL, ERIK GERHARD
Degree: PH.D.
Year: 1995
Corporate Source/Institution: UNIVERSITY OF MARYLAND COLLEGE PARK (0117)
Chairman: HANAN SAMET
Source: VOLUME 56/11-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 6228. 318 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

This dissertation focuses on spatial data structures and algorithms in the sequential and data-parallel environments. The research on spatial data structures addressed three primary topics; first, four widely used and important spatial data structures (the PMR quadtree, the R- **tree** , the R$\sp+$- **tree** , and the R*- **tree**) were compared in the sequential environment using very large, real-world line segment datasets and a varied collection of spatial queries. The collection of spatial queries was intended to represent those that are most commonly used with spatial databases. The goal was not to determine which is the superior representation; instead, it was to demonstrate the various differences and performance tradeoffs that exist between them. The second goal was to extend the spatial data structures into the data-parallel domain and develop data-parallel spatial query algorithms. We defined and implemented a data-parallel PMR quadtree, a data-parallel R- **tree** , and a data-parallel R$\sp+$- **tree** . We developed a collection of spatial queries that are appropriate in comparing the performance of the structures in the data-parallel environment. For example, we investigated different approaches of implementing the data-parallel spatial join algorithm, exploring both top-down and bottom-up algorithmic approaches. The third and final goal was to compare the performance of the the spatial structures in the data-parallel environment in a similar manner as was done in the sequential domain. In addition to measuring raw algorithmic performance, our experiments also focused on the effects of varying the fundamental data structure parameters such as splitting thresholds and **node / bucket** capacities.

22/5/14 (Item 2 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01448732 ORDER NO: AADAA-I9540376
NEW CLASSIFICATION, CODEBOOK DESIGN AND ENCODING ALGORITHMS FOR VECTOR QUANTIZATION OF IMAGES (DATA COMPRESSION)
Author: QUWEIDER, MAHMOUD KH.
Degree: PH.D.
Year: 1995
Corporate Source/Institution: THE UNIVERSITY OF TOLEDO (0232)
Adviser: E. SALARI
Source: VOLUME 56/07-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 3936. 140 PAGES
Descriptors: ENGINEERING, ELECTRONICS AND ELECTRICAL ; COMPUTER SCIENCE
Descriptor Codes: 0544; 0984

Vector quantization (VQ) has proven to be an effective method for high ratio data compression. However, the computational complexity of the encoding process as well as the edge degradation at low bit rates have

limited its use. Classified VQ (CVQ) **tries** to combat these problems by classifying the input block into one of M classes prior to its encoding. The aim of this research is three fold: (1) to develop new classification techniques that are easy to implement in software and hardware; (2) to devise fast, effective and systematic ways to populate the codebooks; (3) to create new fast encoding algorithms to combat the computational complexity associated with VQ.

Two classification algorithms were developed: local thresholding which works in the spatial domain and Peano scanning-based classifier which works both in the transform and spatial domains. The first algorithm binarizes a block based on a local threshold. It then compares the binary block with a predefined set of binary edge templates to decide its class. The algorithm is simple and easy to implement in parallel using array processors with little **memory** requirements. The second algorithm uses the clustering properties of Peano scanning to create a look up **table** used for classification. Determining the class of the input requires logarithmic time proportional to the number of entries in the **table** .

For efficient and systematic bit allocation among the M classes, single **tree** growing was extended to M **trees** . The growing is accomplished simultaneously in an interconnected fashion one node at a time. The terminal nodes of each **tree** constitute the final codebook of its class. A second algorithm based on the ac energy measure is also developed. It incorporates the classification process to design a mean-removed codebook for each class. The codebook is generated by averaging vectors falling in a given interval of the ac energy range.

For encoding, a fast approximate encoding algorithm, that comes within a fraction of a dB of the minimum mean square full search, based on the Peano scanning clustering properties was developed. It searches a window of vectors rather than a whole codebook. The window is centered around a set of vectors which have the closest Peano scanning, of a feature vector, to that of the input. If exact full-search equivalent encoding is desired, the search can extend outside the window in an up and down manner in association with the partial distance search (PDS) method. This helps to expedite the PDS by finding close matching vectors to the input as soon as possible.

The simulation results for the above algorithms show that high quality images with no edge degradation and high peak-signal-to-noise ratio (PSNR) values are achieved at rates in the range $0.50\sim0.82$ bits per pixel (bpp), with greatly reduced computational and implementation complexity.

22/5/15 (Item 3 from file: 35)

01440474 ORDER NO: AADAA-I9535333

COMPRESSED TRIE **VARIATIONS THAT MINIMIZE STORAGE (DATA RETRIEVAL)**

Author: GLANDER, KARL WILLIAM

Degree: PH.D.
Year: 1995
Corporate Source/Institution: COLORADO STATE UNIVERSITY (0053)
Source: VOLUME 56/06-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 3288. 406 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

The basic **trie** structure has the desirable feature of O($\vert$W$\vert$) access time to determine if key word W is contained in the **trie** but **tries** also have the undesirable feature of an excessive amount of wasted storage. Many techniques have been proposed for reducing the storage requirement of **tries** while maintaining the access time. In applications where the data set stored in the **trie** is static, the technique of **trie** compression does well to reduce the **trie** storage requirements. **Trie** compression, however, does not significantly reduce the **trie** storage requirements below that of the commonly used B-tree structure to warrant the time needed to perform the **trie** compression.

This research examines the performance on large static data sets of sixteen **trie** variations that apply a divide and conquer approach to

minimize storage requirements through the efficient storage of the top nodes of the basic **trie** in one data structure and the storage of the subtries that contain the remaining **trie nodes** as **compressed tries** . Data set sizes for this research range from 10,000 to 351,644 key words from a database that contains English words as well as from a database that contains 9-digit numbers.

22/5/16 (Item 4 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01327403 ORDER NO: AADNN-81138
SKIP LISTS AND PROBABILISTIC ANALYSIS OF ALGORITHMS
Author: PAPADAKIS, THOMAS
Degree: PH.D.
Year: 1993
Corporate Source/Institution: UNIVERSITY OF WATERLOO (CANADA) (1141)
Source: VOLUME 54/08-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 4257. 161 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984
ISBN: 0-315-81138-2

This thesis is concerned with various forms of skip lists, and with probabilistic analyses of algorithms. We investigate three topics; one topic from each of these two areas, and another topic common to both of them.

First, we consider Pugh's skip list. We derive exact and asymptotic expressions for the average search cost of a fixed key and of an average key. Our results improve previously known upper bounds of these two average search costs. We also derive exact and asymptotic expressions for the variance of the search cost for the largest key.

Next, we propose several versions of deterministic skip lists. They all have guaranteed logarithmic search and update costs per operation, they lead to an interesting "bridge" structure between the original skip list and standard search **trees** , they are simpler to implement than standard balanced search **trees** , and our experimental results suggest that they are also competitive in terms of space and time.

Finally, we consider the elastic-budget trie, a variant of the standard trie, in which each external **node** (**bucket**) has precisely as many key slots as the number of keys stored in it. We examine the number of buckets of each size, and we derive exact and asymptotic expressions for their average values, as well as asymptotic expressions for their variances and covariances under the closely related "Poisson model" of randomness. Our experimental results suggest that maintaining only two bucket sizes may be a very reasonable practical choice.

22/5/17 (Item 1 from file: 202)
DIALOG(R)File 202:Info. Sci. & Tech. Abs.

3002687
Key-range locking with index trees .
Author(s): Green, R J; Lomet, D.B.
Patent Number(s): US 5440732
Publication Date: Aug 8, 1995
Language: English
Document Type: Patent
Record Type: Abstract
Journal Announcement: 3000

A database-management system generates bounded-disorder indexes on its database keys. In such an index, the leaf nodes are large and are divided into a number of buckets, only one of which ordinarily is accessed in any given single-record database operation. The key values in a leaf node are distributed among the leaf **node** 's **buckets** in accordance with a hashing

function. The lockable ranges locked for scanning functions are defined in accordance with key-valued locking, in which each lockable range is bounded by successive key values that exist in the database. But the multiple-bucket accesses that would otherwise be required, because of the hash-function distribution of key values among a **node** 's several **buckets** , are avoided because the lockable ranges are defined by the sequence of key values in the bucket rather than in the node. In addition to the existing key values, moreover, the bucket's key-value limits are also employed to bound lockable ranges, even if no database records contain those key-value limits. This prevents end-of-bucket insertions and deletions from needing further I/O operations in order to identify the lockable ranges that those insertions and deletions modify.

Descriptors: Access; Database management systems; Hashing; Indexes
Classification Codes and Description: 5.06 (Software and Programming); 6.02 (Bibliographic Search Services, Databases)
Main Heading: Information Processing and Control; Information Systems and Applications

22/5/18 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC

5853870 INSPEC Abstract Number: A9808-3380G-002
Title: Classical and quantum dynamics of chirped pulse dissociation of diatomic molecules
Author(s): Yuan, J.-M.; Wing-Ki Liu
Author Affiliation: Dept. of Phys. & Atmos. Sci., Drexel Univ., Philadelphia, PA, USA
Journal: Physical Review A (Atomic, Molecular, and Optical Physics) vol.57, no.3 p.1992-2001
Publisher: APS through AIP,
Publication Date: March 1998 Country of Publication: USA
CODEN: PLRAAN ISSN: 1050-2947
SICI: 1050-2947(199803)57:3L.1992:CQDC;1-G
Material Identity Number: N687-98003
U.S. Copyright Clearance Center Code: 1050-2947/98/57(3)/1992(10)/$15.00
Document Number: S1050-2947(98)03403-9
Language: English Document Type: Journal Paper (JP)
Treatment: Theoretical (T)
Abstract: The dissociation of a diatomic molecule by a chirped infrared laser pulse is modeled by a Morse oscillator interacting with a classical electric field with a time-dependent frequency. Our previous classical analysis in terms of bucket dynamics, in which systems within the single-**node** separatrices (**buckets**) in phase space are trapped and undergo convection to highly excited states, is found to be applicable to the more general cases of nonlinear chirping and using a realistic dipole moment function for the molecule. This route of excitation leads to a much lower dissociation threshold laser intensity when compared to the chaotic diffusion route for monochromatic excitation. Time-dependent quantum mechanical calculations of the dissociation probability based on the split-operator method are performed. It is found that the classical and quantum results agree well, and the classical resonances appear also in the quantum probabilities. Hence the classical method can be used to investigate various characteristics of the chirped pulse excitation and dissociation processes. (41 Refs)
Subfile: A
Descriptors: chirp modulation; classical mechanics; excited states; molecular moments; Morse potential; optical modulation; photodissociation; quantum theory
Identifiers: chirped pulse dissociation; diatomic molecules; quantum dynamics; classical dynamics; chirped infrared laser pulse; Morse oscillator; classical electric field; time-dependent frequency; classical analysis; bucket dynamics; single-node separatrices; phase space; convection; highly excited states; nonlinear chirping; realistic dipole moment function; dissociation threshold; chaotic diffusion route; monochromatic excitation; time-dependent quantum mechanical calculations;

dissociation probability; split-operator method; classical resonances; chirped pulse excitation
Class Codes: A3380G (Diffuse molecular spectra; predissociation, photodissociation); A8250F (Photolysis and photodissociation by IR, UV and visible radiation); A3420 (Interatomic and intermolecular potentials and forces); A3150 (Excited states of atoms and molecules)

22/5/19 (Item 2 from file: 2)
DIALOG(R)File 2:INSPEC

5084410 INSPEC Abstract Number: A9522-3380G-011
Title: Nonlinear dynamics of chirped pulse excitation and dissociation of diatomic molecules
Author(s): Wig-Ki Liu; Binruo Wu; Jian-Min Yuan
Author Affiliation: Dept. of Phys., Waterloo Univ., Ont., Canada
Journal: Physical Review Letters vol.75, no.7 p.1292-5
Publication Date: 14 Aug. 1995 Country of Publication: USA
CODEN: PRLTAO ISSN: 0031-9007
U.S. Copyright Clearance Center Code: 0031-9007/95/75(7)/1292(4)$06.00
Language: English Document Type: Journal Paper (JP)
Treatment: Theoretical (T)
Abstract: The classical dynamics of a diatomic molecule modeled by a Morse oscillator interacting with a chirped infrared laser pulse is studied. When the chirping rate is small, the system can be described approximately in the moving frame by a time-independent Hamiltonian, which produces single- **node** separtrices (**buckets**) in phase space. Systems trapped in the buckets undergo convection to dissociation. This route to dissociation is different from the chaotic diffusion route for monochromatic excitation and requires a much lower threshold laser intensity to achieve dissociation. (14 Refs)
Subfile: A
Descriptors: molecule-photon collisions; photodissociation; photoexcitation
Identifiers: nonlinear dynamics; excitation; dissociation; diatomic molecules; classical dynamics; Morse oscillator; chirped infrared laser pulse; time-independent Hamiltonian; single-node separtrices; phase space; chaotic diffusion route
Class Codes: A3380G (Diffuse molecular spectra; predissociation, photodissociation); A8250F (Photolysis and photodissociation by IR, UV and visible radiation)

22/5/20 (Item 3 from file: 2)
DIALOG(R)File 2:INSPEC

4676170 INSPEC Abstract Number: A9413-0320-002
Title: Particle dynamics in chirped-frequency fluctuations
Author(s): Hsu, C.T.; Cheng, C.Z.; Helander, P.; Sigmar, D.J.; White, R.
Author Affiliation: Plasma Fusion Center, MIT, Cambridge, MA, USA
Journal: Physical Review Letters vol.72, no.16 p.2503-7
Publication Date: 18 April 1994 Country of Publication: USA
CODEN: PRLTAO ISSN: 0031-9007
U.S. Copyright Clearance Center Code: 0031-9007/94/72(16)/2503(5)$06.00
Language: English Document Type: Journal Paper (JP)
Treatment: Theoretical (T)
Abstract: Hamiltonian systems describing particle motion in a wave with time-dependent (chirped) frequency are studied. The wave is found to form a single- **node** separatrix (**bucket**) moving in the phase space at a rate proportional to that of the frequency change. Particles trapped inside the bucket undergo convection, while untrapped particles colliding with the bucket get a resonant kick, in phase space. In toroidal systems, these effects can result in a large radial convective flux roughly proportional to the size of the bucket and the frequency chirping. Possible applications

of this novel mechanism to tokamak plasmas are discussed. (15 Refs)
Subfile: A
Descriptors: dynamics; fluctuations; plasma kinetic theory
Identifiers: chirped-frequency fluctuations; Hamiltonian systems; particle motion; wave; single-node separatrix; phase space; convection; untrapped particles; trapped particles; toroidal systems; large radial convective flux; tokamak plasmas; particle dynamics
Class Codes: A0320 (Classical mechanics of discrete systems: general mathematical aspects); A5220D (Particle orbits); A5225D (Plasma kinetic equations)

22/5/21 (Item 4 from file: 2)
DIALOG(R)File 2:INSPEC

04297596 INSPEC Abstract Number: C9301-6120-025
Title: Fragmentary string matching by selective access to hybrid tries
Author(s): Dengel, A.; Pleyer, A.; Hoch, R.
Author Affiliation: German Res. Center for Artificial Intelligence, DFKI, Kaiserslautern, Germany
Conference Title: Proceedings. 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems p.149-53
Publisher: IEEE Comput. Soc. Press, Los Alamitos, CA, USA
Publication Date: 1992 Country of Publication: USA xxxiv+735 pp.
ISBN: 0 8186 2915 0
U.S. Copyright Clearance Center Code: 0 8186 2915 0/92$3.00
Conference Sponsor: Int. Assoc. Pattern Recognition
Conference Date: 30 Aug.-3 Sept. 1992 Conference Location: The Hague, Netherlands
Language: English Document Type: Conference Paper (PA)
Treatment: Theoretical (T)
Abstract: The authors propose a dictionary look-up method as a contextual postprocessing for character hypotheses forming word candidates. In particular, a hybrid **trie** organization is combined with a selective-access-matrix (SAM) that allows an efficient matching of fragmentary input strings against legal words. Experiments prove that the method achieves some respectable results concerning speed. Furthermore, the additional **memory** needed for the SAM is smaller than the **memory** saved by the hybrid organization of the **trie** . (15 Refs)
Subfile: C
Descriptors: document image processing; optical character recognition; **table** lookup; **trees** (mathematics)
Identifiers: hybrid **trie** selective access; fragmentary string matching; dictionary look-up method; contextual postprocessing; character hypotheses; word candidates; hybrid **trie** organization; selective-access-matrix
Class Codes: C6120 (File organisation); C6130D (Document processing techniques); C5260B (Computer vision and picture processing); C1250B (Character recognition)

22/5/22 (Item 5 from file: 2)
DIALOG(R)File 2:INSPEC

03966173 INSPEC Abstract Number: C91058959
Title: Improved display algorithm for linear octrees
Author(s): Yang, S.N.; Jong, B.S.
Author Affiliation: Inst. of Comput. Sci., Nat. Tsing Hua Univ., Hsinchu, Taiwan
Conference Title: Proceedings of the 2nd International Conference. Pixim 89. Computer Graphics in Paris p.435-17
Publisher: Hermes, Paris, France
Publication Date: 1989 Country of Publication: France xvi+525 pp.
ISBN: 2 86601 196 1
Conference Date: 25-29 Sept. 1989 Conference Location: Paris, France
Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P); Theoretical (T)
Abstract: An improved algorithm for displaying linear octree encoded solids is given. The underlying notion is based on finding a hash function which will map octree **nodes** into **buckets** arranged in a back-to-front or front-to-back display sequence with respect to the given viewing direction. The algorithm is optimal in the sense that its time complexity is proportional to the number of black nodes. That is, it runs in O(B) time which is better than the existing algorithm with (nB) time complexity, where B is the number of black nodes and n is the resolution parameter. (22 Refs)
Subfile: C
Descriptors: computational complexity; computational geometry; computer graphics; **trees** (mathematics)
Identifiers: hidden surfaces; computer graphics; linear octrees; hash function; buckets; display sequence; time complexity; black nodes
Class Codes: C6130B (Graphics techniques); C4240 (Programming and algorithm theory); C1160 (Combinatorial mathematics)

22/5/23 (Item 6 from file: 2)
DIALOG(R)File 2:INSPEC

03885795 INSPEC Abstract Number: C91036456
Title: Locally balanced compact trie **hashing**
Author(s): Otoo, E.J.
Author Affiliation: Sch. of Comput. Sci., Carleton Univ., Ottawa, Ont., Canada
Conference Title: Proceedings of the Third International Conference on Data and Knowledge Bases: Improving Usability and Responsiveness p. 242-54
Editor(s): Beeri, C.; Schmidt, J.W.; Dayal, U.
Publisher: Morgan Kauffmann, San Matheo, CA, USA
Publication Date: 1988 Country of Publication: USA vii+424 pp.
Conference Date: 28-30 June 1988 Conference Location: Jerusalem, Israel
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: The author presents an approach to implementing **trie** hashing that revolves the problem of potential degeneracy. Given that the **trie** associated with **trie** hashing is **memory** resident, this approach still retains the virtues that it requires only one disk access for a record lookup and that the scheme is order preserving thus facilitating range retrievals. The additional features of the approach over the original scheme are that: (i) the **trie** is locally height balance, (ii) the digit numbers along any path from the root to a leaf node are always non-decreasing, (iii) the digit values along any single path are compacted into a single node, and (iv) the storage utilization of the data **buckets** is guaranteed to be at least 50%. This approach considerably improves the performance characteristics of the **trie** hashing scheme. (29 Refs)
Subfile: C
Descriptors: data structures; file organisation; **trees** (mathematics)
Identifiers: file organisation; compact **trie** hashing; **memory** resident ; disk access; record lookup; order preserving; range retrievals; locally height balance; storage utilization; data **buckets** ; performance characteristics
Class Codes: C6120 (File organisation)

22/5/24 (Item 1 from file: 94)
DIALOG(R)File 94:JICST-EPlus

04001020 JICST ACCESSION NUMBER: 99A0204520 FILE SEGMENT: JICST-E
Proposal of CDV control method for the transmission of teleproction signals applying the leaky bucket algorithm.
FUJIKAWA FUYUKI (1)
(1) Denryoikuchuken Johoken

Denryoku Chuo Kenkyujo Joho Kenkyujo Hokoku, 1998, NO.R97021, PAGE.18P, FIG.26, TBL.1, REF.4
JOURNAL NUMBER: L1889ABA
UNIVERSAL DECIMAL CLASSIFICATION: 621.311.1
LANGUAGE: Japanese COUNTRY OF PUBLICATION: Japan
DOCUMENT TYPE: Technical Report
ARTICLE TYPE: Original paper
MEDIA TYPE: Printed Publication
ABSTRACT: We proposed the methods to transmit teleplotection signals multiplexed with other traffics within the allowable transmission delay. First proposal is the methodof reducing CDV(cell delay variation) of teleprotection signals and how to determine the shaping value of the traffic which is multiplexed with teleprotection signals. We confirm that our proposal is available for transmitting teleprotection signals. But it is difficult to determine the shaping value of the various traffics. So, we proposed second method to control dynamically CDV of those signals. The proposal method associated with monitoring output of ATM switching node, polices the threshold of CDV and shapes the cell streams except teleprotection signals at input of its **node** , applying the leaky **bucket** algorithm. We can handle teleprotection signals multiplexed by various traffics keeping the allowable delay variation. If we equip an ATM switching node with our proposed mechanism for both outcoming line and incoming line, setting the same threshold of CDV, we may cancel the differential delay time between forward line and backward line. (author abst.)
DESCRIPTORS: electric power system; communication network; ATM network; jitter; flow control(information); delay characteristic; tolerance limit; time limit; traffic monitoring; threshold
BROADER DESCRIPTORS: system; information network; network; electric fluctuation; fluctuation and variation; control; transmission characteristic; characteristic; limit; communication monitoring; monitoring; communication administration; management; numerical value
CLASSIFICATION CODE(S): NB02000E

22/5/27 (Item 1 from file: 34)
DIALOG(R)File 34:SciSearch(R) Cited Ref Sci

02428982 Genuine Article#: LA716 Number of References: 12
Title: SINGLE ACCESS HASHING WITH OVERFLOW SEPARATORS FOR DYNAMIC FILES
Author(s): CESARINI F; SODA G
Corporate Source: UNIV FLORENCE,DIPARTIMENTO SISTEMI & INFORMAT/I-50139 FLORENCE//ITALY/
Journal: BIT, 1993, V33, N1, P15-28
ISSN: 0006-3835
Language: ENGLISH Document Type: ARTICLE
Geographic Location: ITALY
Subfile: SciSearch
Journal Subject Category: COMPUTER APPLICATIONS & CYBERNETICS
Abstract: The dynamic external hashing proposed in this paper allocates records according to the spiral storage technique. Separators derived from the signature technique are used for distinguishing primary from overflow records and for subdividing overflow chains into segments allocated into the primary file. Single access retrieval is obtained by means of a main **memory** index with an entry per **bucket** and containing separators and pointers. While this method uses a larger index than other recent proposals, it is much more convenient regarding load factor and insertion cost. Furthermore, file expansion is directed by various control parameters, thus allowing the user to choose the most suitable policy for his application.
Descriptors--Author Keywords: HASHING ; SPIRAL STORAGE ; SINGLE ACCESS RETRIEVAL ; MAIN **MEMORY** INDEX ; OVERFLOW SEPARATORS
Identifiers--KeyWords Plus: ORGANIZATION; RETRIEVAL
Research Fronts: 91-2055 001 (SPATIAL DATABASES; BINARY SEARCH **TREE** ; DATA ACCESS; ADAPTIVE HASHING; LINEAR OCTREE; GEOGRAPHIC INFORMATION-SYSTEMS; DIGITAL **TRIES**)
Cited References:

CESARINI F, 1991, V16, P309, ACM T DATABASE SYST
ENBODY RJ, 1988, V20, P85, COMPUT SURV
FAGIN R, 1979, V4, P315, ACM T DATABASE SYSTE
GONNET GH, 1988, V35, P161, J ACM
GONNET GH, 1982, P256, P ACM S PRINCIPLES D
KJELBERG P, 1984, P481, 10TH P INT C VER LAR
LARSON PA, 1988, V13, P366, ACM T DATABASE SYST
LARSON PA, 1978, V18, P184, BIT
LARSON PA, 1984, V27, P670, COMMUN ACM
MULLIN JK, 1985, V28, P330, COMPUT J
RAMAKRISHNA MV, 1989, V14, P231, ACM T DATABASE SYST
RAMAKRISHNA MV, 1989, P187, P FODO89 PARIS

22/5/28 (Item 2 from file: 34)
DIALOG(R)File 34:SciSearch(R) Cited Ref Sci

02258449 Genuine Article#: KN982 Number of References: 12
Title: MULTI- DIRECTORY **HASHING**
Author(s): PRAMANIK S; ANALYTI A; DAVIES H; CHOU HY
Corporate Source: MICHIGAN STATE UNIV,DEPT COMP SCI/E LANSING//MI/48824
Journal: INFORMATION SYSTEMS, 1993, V18, N1 (JAN), P63-74
ISSN: 0306-4379
Language: ENGLISH Document Type: ARTICLE
Geographic Location: USA
Subfile: SciSearch; CC ENGI--Current Contents, Engineering, Technology & Applied Sciences
Journal Subject Category: COMPUTER APPLICATIONS & CYBERNETICS
Abstract: We present a new dynamic hashing scheme for disk-based databases, called Multi- **Directory** Hashing (MDH). MDH uses multiple hash **directories** to access a file. The size of each hash **directory** grows dynamically with the file size. The advantages of MDH are enhanced concurrency, improved **bucket** utilization and smaller total **directory** size than single- **directory** hashing. The expected utilization of MDH increases monotonically and approaches 100% as the number of hash **directories** increases. A variation of MDH, called Main **Memory** Multi- **Directory** Hashing (MM-MDH), is also described. MM-MDH achieves optimal search time in main **memory** databases. The performance of both methods is analyzed through theoretical and experimental results.
Descriptors--Author Keywords: MULTI- **DIRECTORY** HASHING ; EXTENDIBLE HASHING ; PARALLEL PROCESSING ; MAIN **MEMORY** DATABASES ; PERFORMANCE ANALYSIS
Research Fronts: 91-2055 002 (SPATIAL DATABASES; BINARY SEARCH **TREE** ; DATA ACCESS; ADAPTIVE HASHING; LINEAR OCTREE; GEOGRAPHIC INFORMATION-SYSTEMS; DIGITAL **TRIES**)
Cited References:
ENBODY RJ, 1988, V20, P85, COMPUT SURV
FAGIN R, 1979, V4, P315, ACM T DATABASE SYSTE
FRAJOLET P, 1983, V20, P345, ACTA INFORM
GARY AK, 1986, V11, P213, ACM T DATABASE SYST
LARSON P, 1988, V31, P446, COMMUN ACM
LARSON P, 1980, P224, 6TH P C VER LARG DAT
LARSON PA, 1982, V7, P566, ACM T DATABASE SYST
LARSON PA, 1978, V18, P184, BIT
LEHMAN TJ, 1986, P294, 12TH P VLDB C
LITWIN W, 1980, P212, 6TH P INT C VER LARG
SEVERANCE C, 1990, P674, 16TH P VLDB C
WIEDERHOLD G, 1987, FILE ORG DATABASE DE

22/5/29 (Item 1 from file: 95)
DIALOG(R)File 95:TEME-Technology & Management

01051529 I96120007259
Approximate average storage utilization of bucket **methods with arbitrary fanout**

(Genaeherte mittlere Speicherausnutzung bei der **Bucket** -Methode)
Chuan-Heng Ang; Samet, H
Dept. of Inf. Syst. & Comput. Sci., Nat. Univ. of Singapore, Singapore
Nordic Journal of Computing, v3, n3, pp280-291, 1996
Document type: journal article Language: English
Record type: Abstract
ISSN: 1236-6064

ABSTRACT:
The approximate average storage utilization of **bucket** methods with fanout of n, assuming a uniform distribution of data, is shown to depend only on the fanout and not on any other factors such as the **directory** structure. It obeys the formula (ln n)/(n-1). The analysis makes use of a generalization of previously published methods for n=2 and n=4 and its predictions match these results. The formula is applicable to methods that are based on digital searching (e.g., **tries**) or balancing rather than comparison based methods. The formula is also used to detect an erroneous statement about the average storage utilization of a ternary system by J. Nievergelt et al. (1984).

DESCRIPTORS: FILE MANAGEMENT; **MEMORY** MANAGEMENT; SEARCH ALGORITHM; **TREE** STRUCTURE; DATA MODELS
IDENTIFIERS: APPROXIMATE AVERAGE STORAGE UTILIZATION; **BUCKET** METHODS; ARBITRARY FANOUT; UNIFORM DISTRIBUTION; **DIRECTORY** STRUCTURE; DIGITAL SEARCHING; **TRIES** ; ERRONEOUS STATEMENT; AVERAGE STORAGE UTILIZATION; Dateiverwaltung; mittlere Speicherausnutzung

25/5/1 (Item 1 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

06576507 E.I. No: EIP03437689548
Title: An Experimental Study of Compression Methods for Dynamic Tries
Author: Nilsson, S.; **Tikkanen, M.**
Corporate Source: KTH, Nada, SE-100 44 Stockholm, Sweden
Source: Algorithmica (New York) v 33 n 1 SPEC.ISS. May 2002. p 19-33
Publication Year: 2002
CODEN: ALGOEJ ISSN: 0178-4617
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 0310W4
Abstract: We study an order-preserving general purpose data structure for binary data, the LPC- **trie** . The structure is a compressed **trie** , using both level and path compression. The memory usage is similar to that of a balanced binary search tree, but the expected average depth is smaller. The LPC- **trie** is well suited to modern language environments with efficient memory allocation and garbage collection. We present an implementation in the Java programming language and show that the structure compares favorably with a balanced binary search tree. 27 Refs.
Descriptors: *Data structures; Java programming language; Trees (mathematics); Problem solving
Identifiers: Binary data
Classification Codes:
723.1.1 (Computer Programming Languages)
723.2 (Data Processing); 723.3 (Database Systems); 723.1 (Computer Programming); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 723.4 (Artificial Intelligence)
723 (Computer Software, Data Handling & Applications); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

25/5/2 (Item 1 from file: 65)
DIALOG(R)File 65:Inside Conferences

03522523 INSIDE CONFERENCE ITEM ID: CN037120315
Implementing a dynamic compressed trie
Nilsson, S.; **Tikkanen, M.**
CONFERENCE: Workshop on algorithm engineering; WAE'98-2nd
MAX PLANCK INSTITUT FUR INFORMATIK, 1998; (NO) 1-019 P: 25-36
MPI Informatik, 1998
ISSN: 0946-011X
LANGUAGE: English DOCUMENT TYPE: Conference Papers
CONFERENCE EDITOR(S): Mehlhorn, K.
CONFERENCE SPONSOR: Max-Planck-Institut fur Informatik
CONFERENCE LOCATION: Saarbrucken, Germany 1998; Aug (199808) (199808)

BRITISH LIBRARY ITEM LOCATION: 5413.280870
DESCRIPTORS: algorithm engineering; WAE; informatik

File 348:EUROPEAN PATENTS 1978-2004/Sep W02
(c) 2004 European Patent Office
File 349:PCT FULLTEXT 1979-2002/UB=20040916,UT=20040909
(c) 2004 WIPO/Univentio

Set	Items	Description
S1	339504	MEMOR??? OR RAM OR DRAM OR SRAM OR SDRAM OR RDRAM OR SLDRAM OR SGRAM OR DRDRAM OR ROM OR PROM OR EPROM OR EEPROM OR FPO - OR EDO
S2	76569	DIRECTORY OR DIRECTORIES OR HIERARCH? OR TREE? ?
S3	21	TRIE()NODE? ?
S4	506984	TABLE? ? OR LUT? ?
S5	137	(POINT??? OR ADDRESS???)(5N)((LOWER OR DEEPER)(3N)NODE? ?)
S6	1205	BUCKET? ?(10N)(DATA OR INFORMATION OR POINT??? OR ADDRESS?-?? OR S2)
S7	10	S1(50N)S2(50N)TRIE? ?(50N)BUCKET? ?
S8	8	S1(50N)S2(50N)S3
S9	121	S1(50N)S2(50N)TRIE? ?
S10	564	TRIE? ?(10N)S4:S5
S11	21	S1(50N)S2(50N)S10
S12	34	S1(50N)S2(50N)S6
S13	53	S7:S8 OR S11:S12
S14	40	S13 AND AC=US/PR
S15	21	S14 AND AY=(1970:1999)/PR
S16	11	S13 AND PY=1970:1998
S17	25	S15:S16

17/3,K/1 (Item 1 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01311092
Route lookup engine
Wege-Nachschlage-Motor
Moteur de recherche dans une table de routage
PATENT ASSIGNEE:
ASCEND COMMUNICATIONS, INC., (1470333), One Ascend Plaza, 1701 Harbor Bay Parkway, Alameda, CA 94502, (US), (Applicant designated States: all)
INVENTOR:
Hebb, Andrew T., 62 Lakeside Avenue, Hudson, Massachusetts 01749, (US)
Cherian, Sanjay G., 6 Maxwell Drive, Brookline, New Hampshire 03033, (US)
LEGAL REPRESENTATIVE:
Watts, Christopher Malcolm Kelway, Dr. (37391), Lucent Technologies (UK) Ltd, 5 Mornington Road, Woodford Green Essex, IG8 0TU, (GB)
PATENT (CC, No, Kind, Date): EP 1122927 A2 010808 (Basic)
EP 1122927 A3 040414
APPLICATION (CC, No, Date): EP 2000310758 001204;
PRIORITY (CC, No, Date): US 459441 991213
DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE; TR
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: H04L-029/06; G06F-017/30; H04L-012/56; H04Q-011/04
ABSTRACT WORD COUNT: 99
NOTE:
Figure number on first page: 3

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200132	908
SPEC A	(English)	200132	3089
Total word count - document A			3997
Total word count - document B			0
Total word count - documents A + B			3997

...SPECIFICATION and subsequently on the IP source address if the packet is a multicast packet. The search utilizes a multi-bit **tree** search with prefix expansion and capture. The search terminates when a next-hop index is found or the end of...

...header search at the end of the IP source address. An RLE manager of the routing subsystem manages the RLE **memory** . The RLE **memory** is used to store the hardware Variable Stride **Trie** (VST) route **tables** for each of the configured Virtual Private Networks (VPNs) including the default VPN, VPNO. Each hardware VST is guaranteed a...

17/3,K/2 (Item 2 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01277549
PATTERN MATCHING FOR DATA EXCHANGE BETWEEN COMPUTER AIDED DESIGN SYSTEMS
MUSTERERKENNUNG FUR DEN DATENAUSTAUSCH ZWISCHEN RECHNERGESTUTZTEN ENTWURFSSYSTEMEN
FILTRAGE DESTINE A L'ECHANGE DE DONNEES ENTRE DES SYSTEMES DE CONCEPTION ASSISTEE PAR ORDINATEUR
PATENT ASSIGNEE:
Proficiency Ltd, (3273762), 8 HaMarpeh Street, 91450 Jerusalem, (IL), (Proprietor designated states: all)
INVENTOR:
RAPPOPORT, Ari, 8 HaMarpeh Street, 91450 Jerusalem, (IL)
LEGAL REPRESENTATIVE:
Viering, Jentschura & Partner (100645), Postfach 22 14 43, 80504 Munchen,

(DE)
PATENT (CC, No, Kind, Date): EP 1226514 A2 020731 (Basic)
EP 1226514 B1 040317
WO 2001018672 010315
APPLICATION (CC, No, Date): EP 2000962769 000906; WO 2000IB1439 000906
PRIORITY (CC, No, Date): US 391311 990907
DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-017/50
NOTE:
No A-document published by EPO
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200412	734
CLAIMS B	(German)	200412	689
CLAIMS B	(French)	200412	851
SPEC B	(English)	200412	10937
Total word count - document A			0
Total word count - document B			13211
Total word count - documents A + B			13211

...SPECIFICATION Additional data structures can also be included. For instance, as was mentioned above with reference to FIG. 6, the match **data** records 709 can be broken into various hash **buckets** by using known hashing techniques, or a B- **tree** or other type of indexing structure can be used to expedite search operations. Moreover, it can be efficient to sort...
...709 prior to run-time or once the records have been updated. If the records are sorted then regions of **memory** with a high locality of reference (meaning that if a **memory** address X is called, then **memory** address Y is likely to be called too) can be grouped together, thereby reducing I/O and read latencies.
FIG...

17/3,K/5 (Item 5 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01277089
METHOD AND APPARATUS FOR EDGE CORRELATION BETWEEN DESIGN OBJECTS
VERFAHREN UND GERAT ZUR KANTENKORRELATION ZWISCHEN DESIGN-OBJEKTEN
PROCEDE ET APPAREIL DESTINES A LA CORRELATION DE CONTOURS D'OBJETS MODELISES
PATENT ASSIGNEE:
Proficiency Ltd, (3273762), 8 HaMarpeh Street, 91450 Jerusalem, (IL), (Proprietor designated states: all)
INVENTOR:
ETZION, Michal, 8 HaMarpeh Street, 91450 Jerusalem, (IL)
SPITZ, Steven, 8 HaMarpeh Street, 91450 Jerusalem, (IL)
LEGAL REPRESENTATIVE:
Viering, Jentschura & Partner (100645), Postfach 22 14 43, 80504 Munchen, (DE)
PATENT (CC, No, Kind, Date): EP 1218828 A2 020703 (Basic)
EP 1218828 B1 030820
WO 2001018669 010315
APPLICATION (CC, No, Date): EP 2000958951 000906; WO 2000IB1355 000906
PRIORITY (CC, No, Date): US 391311 990907
DESIGNATED STATES (Pub A): AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; (Pub B): AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-017/00
NOTE:
No A-document published by EPO
LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200334	742
CLAIMS B	(German)	200334	661
CLAIMS B	(French)	200334	871
SPEC B	(English)	200334	10968
Total word count - document A			0
Total word count - document B			13242
Total word count - documents A + B			13242

...SPECIFICATION Additional data structures can also be included. For instance, as was mentioned above with reference to FIG. 6, the match **data** records 709 can be broken into various hash **buckets** by using known hashing techniques, or a B- **tree** or other type of indexing structure can be used to expedite search operations. Moreover, it can be efficient to sort...
...709 prior to run-time or once the records have been updated. If the records are sorted then regions of **memory** with a high locality of reference (meaning that if a **memory** address X is called, then **memory** address Y is likely to be called too) can be grouped together, thereby reducing I/O and read latencies.
FIG...

17/3,K/6 (Item 6 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01143736
Method and system for manipulating the order soft permanent connections are released
Verfahren und System zur Manipulation der Reihenfolge des Auslosens der "Soft" permanenten Verbindungen
Procede et systeme pou manipuler l'ordre pour terminer des connexions permanents "soft"
PATENT ASSIGNEE:
Fore Systems, Inc., (2032081), 1000 Fore Drive, Warrendale, Pennsylvania 15086, (US), (Applicant designated States: all)
INVENTOR:
Mallath, Harisankar C., 5900 Babcock Blvd., Chapel Hill Apt 43, Pittsburgh, PA 15237-2555, (US)
Santhanakrishnan, Ramprasad, 4511 Haddon Place, Wexford, PA 15090, (US)
LEGAL REPRESENTATIVE:
O'Connell, David Christopher (62551), Haseltine Lake & Co., Imperial House, 15-19 Kingsway, London WC2B 6UD, (GB)
PATENT (CC, No, Kind, Date): EP 998163 A2 000503 (Basic)
APPLICATION (CC, No, Date): EP 99308410 991025;
PRIORITY (CC, No, Date): US 179030 981026
DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: H04Q-011/04
ABSTRACT WORD COUNT: 170
NOTE:
Figure number on first page: 1

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200018	1160
SPEC A	(English)	200018	2774
Total word count - document A			3934
Total word count - document B			0
Total word count - documents A + B			3934

...SPECIFICATION organized in a separate list associated with each service category of the signaling interface in the form of a splay **tree** as shown in figure 4. For a given type of connection of the signaling

interface, as shown in the figure...

...connections are released, they will be released one after the other throughout all the hash buckets. Similarly, all of the **data** connections are linked through the hash buckets so the **data** connections are released in order, regardless of what hash **buckets** they are in. In turn, each splay **tree** of a connection is comprised of splay **trees** based on whether they are CBR, VBR, ABR or UBR connections, as shown in figure 5. The splay **tree** affords the ability to add or remove connections easily, as is well known in the art in regard to this...

...system when there is no failure of the network and connections are being added or removed over time. The splay **tree** is formed of pointer mechanisms, each of which maintains a pointer to a **memory** location having the associated connection information, and a pointer to the next pointer mechanism associated with the next connection in...

...CLAIMS the service category of the connections.

10. A system as described in Claim 9 wherein the list includes a splay **tree** of connections for each service category for the signaling interface, and the manipulating mechanism includes a controller which maintains the splay **tree** for each service category.
11. A system as described in Claim 10 wherein the look up **data** structures include hash **buckets** .
12. A system as described in Claim 11 wherein the manipulating mechanism releases connections having service categories of voice or video before connections having service categories of data and according to their respective splay **tree** .
13. A system as described in Claim 12 wherein the connections includes SPVx connections and each splay **tree** includes all of the SPVx connections for each service category sorted in release priority order.
14. A system as described in Claim 13 wherein the controller sorts the splay **tree** within a service category in the release priority order of sub-category SPVPCs, followed by sub-category SPVCs, followed by ...

...UBR connections.

16. A switch for routing established SPVx connections of a telecommunications network having a telecommunications system comprising:

a **memory** ; and

a list of the connections, said list disposed in said memory; and

a mechanism for manipulating the order in...

17/3,K/7 (Item 7 from file: 348)

DIALOG(R)File 348:EUROPEAN PATENTS

01086103

FAST STRING SEARCHING AND INDEXING

SCHNELLES ZEICHENKETTENSUCHEN UND -INDIZIEREN

RECHERCHE ET INDEXATION RAPIDES DE CHAINES DE CARACTERES

PATENT ASSIGNEE:

SAP Aktiengesellschaft, (2635751), Neurottstrasse 16, 69190 Walldorf, (DE), (Proprietor designated states: all)

INVENTOR:

BRAUN, Bernhard, Jahnweg 6, D-69231 Rauenberg, (DE)

LEGAL REPRESENTATIVE:

Jany, Peter, Dr. (79031), Dr. H.-P. Pfeifer Dr. P. Jany Patentanwalte Beiertheimer Allee 19, 76137 Karlsruhe, (DE)

PATENT (CC, No, Kind, Date): EP 1066570 A1 010110 (Basic)
EP 1066570 B1 031029
WO 99044151 990902

APPLICATION (CC, No, Date): EP 99908946 990225; WO 99EP1210 990225

PRIORITY (CC, No, Date): US 31285 980226

DESIGNATED STATES: AT; BE; CH; DE; FR; GB; IE; IT; LI; LU

INTERNATIONAL PATENT CLASS: G06F-017/30
NOTE:
No A-document published by EPO
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200344	1616
CLAIMS B	(German)	200344	1624
CLAIMS B	(French)	200344	1861
SPEC B	(English)	200344	11304
Total word count - document A			0
Total word count - document B			16405
Total word count - documents A + B			16405

...SPECIFICATION 214 assignment
216 assignment
218 decision
220 termination
222 comparison
224 traversal
230 determination
232 decision
234 decision
240 outer **tree** insert
242 registration
244 insertion
246 insertion
248 decision
260 inner **tree** insert
262 allocation
264 insertion
266 decision
268 registration
270 insertion
272 insertion
280 increment
282 setting
284 return
302 index **tree**
331 pos field
340 hash table array
341 hash **bucket**
350 reference
360 branch
370 link
380 link
402 index **tree**
405 leaf node
408 interior node
440 **memory** manager
460 reference
500 flowchart
504 decision
506 setting
508 removal
510 removal
512 decrement
520 decision
522 decision
524 assignment
526 unlinking
528 insertion
530 removal
540 decision
542 deletion
550 freeing
602 index **tree**
702 index **tree**

710 leaf node
720 hashNext field
750 reference
760 link
800 flowchart
802 reference
804 decision
806 consideration
808 retrieval...

17/3,K/8 (Item 8 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00657595
Efficient storage of object in a file system
Effiziente Speicherung eines Objektes in einem Dateisystem
Stockage efficace d'objet dans un systeme de fichiers
PATENT ASSIGNEE:
MICROSOFT CORPORATION, (749861), One Microsoft Way, Redmond, Washington 98052-6399, (US), (Proprietor designated states: all)
INVENTOR:
Zbikowski, Mark, 15817 N.E. 178th Place, Woodinville, Washington 98072, (US)
Berkowitz, Brian T., 3912-142nd Place N.E., Bellevue, Washington 98007, (US)
Ferguson, Robert I., 2910-9th Avenue West, Seattle, Washington 98119, (US)
LEGAL REPRESENTATIVE:
Grunecker, Kinkeldey, Stockmair & Schwanhausser Anwaltssozietat (100721), Maximilianstrasse 58, 80538 Munchen, (DE)
PATENT (CC, No, Kind, Date): EP 632364 B1 020306 (Basic)
APPLICATION (CC, No, Date): EP 94110003 940628;
PRIORITY (CC, No, Date): US 86344 930630
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: G06F-003/06
ABSTRACT WORD COUNT: 110
NOTE:
Figure number on first page: 1

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF2	2048
CLAIMS B	(English)	200210	1806
CLAIMS B	(German)	200210	1790
CLAIMS B	(French)	200210	2242
SPEC A	(English)	EPABF2	3915
SPEC B	(English)	200210	4106
Total word count - document A			5964
Total word count - document B			9944
Total word count - documents A + B			15908

...CLAIMS B1
1. A method, for use in a data processing system (10) having secondary storage (16) with **memory** space, of storing file data in the secondary storage (16), the method comprising:
logically partitioning at least a portion of the **memory** space in the secondary storage (16) into fixed-sized **data** structures (**Buckets** 1 to N); characterized by
storing a first set of logically contiguous file data in a first variable-sized data...

...among other sets of logically contiguous file data held in the first variable-sized data structure (40);
storing a B- **tree** index (83) of multiple entries in the first variable-sized data structure (40) in the secondary storage (16),

structure (40);
means (12, 24) for storing a B- **tree** index (83) of multiple entries in the first variable-sized data structure (40) in the secondary storage (16), said entries...

...set of logically contiguous file data;
means (12, 24) for storing a second identifier (44) in the first variable-sized **data** structure (40) that uniquely identifies the first variable-sized **data** structure (40) within a fixed-sized **data** structure (**Buckets** 1 to N);
means (12, 24) for storing the first variable-sized data structure (40) in at least one of the fixed-sized **data** structures (**Buckets** 1 to N) in the secondary storage (16); and
means (12, 24) for storing a map (74) of multiple entries...

17/3,K/9 (Item 9 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00593675
Packet format in hub for packet data communications system
Paketformat in Knotenpunkt fur ein Datenubertragungssystem
Format de paquets dans un boitier central pour systeme de communication par paquets
PATENT ASSIGNEE:
CABLETRON SYSTEMS, INC., (1353625), 35 Industrial Way, Rochester, NH 03866, (US), (applicant designated states: DE;FR;GB;IT)
INVENTOR:
Spinney, Barry Alan, 22 Anthony Road, Wayland, MA01778, (US)
Simcoe, Robert J., 1 Brookway Road, Westboro, MA 01581, (US)
Thomas, Robert Eugene, 17 Shawmut Avenue, Hudson, MA 01749, (US)
Varghese, George, 6-F Forest Acres, Bradford, MA 01835, (US)
LEGAL REPRESENTATIVE:
Betten & Resch (101031), Reichenbachstrasse 19, 80469 Munchen, (DE)
PATENT (CC, No, Kind, Date): EP 594199 A1 940427 (Basic)
EP 594199 B1 990707
APPLICATION (CC, No, Date): EP 93117159 931022;
PRIORITY (CC, No, Date): US 965651 921022
DESIGNATED STATES: DE; FR; GB; IT
INTERNATIONAL PATENT CLASS: H04L-012/46;
ABSTRACT WORD COUNT: 181

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9927	862
CLAIMS B	(German)	9927	790
CLAIMS B	(French)	9927	1054
SPEC B	(English)	9927	8079
Total word count - document A			0
Total word count - document B			10785
Total word count - documents A + B			10785

...SPECIFICATION its higher ordered bits, indexed for one-of-22K selection of the word 91 in the table. The translation table **pointer** 93 returned by the hash **bucket** 90 in the hash table 89 is used to select a breadth-first balanced binary **tree** as illustrated in Figure 7. The **trees** are stored in a translation table 94 in **memory** 21, and each **tree** has between one and seven entries, as indicated by the size field 92. The binary **tree** cannot cross a block boundary in the translation table 94 in **memory** 21. The ordering of entries 96 in a breadth-first balanced binary **tree** for various table sizes is illustrated in Figure 7. Note that there are from one to seven entries (each entry...

17/3,K/10 (Item 10 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00593671

Address lookup in packet data communications link, using hashing and content-addressable memory

Aufsuchen von Adressen bei Paketubertragung mittels Hashing und eines inhaltsadressierten Speichers

Selection d'adresses de paquets de communications par hashing et une memoire associative

PATENT ASSIGNEE:
CABLETRON SYSTEMS, INC., (1353625), 35 Industrial Way, Rochester, NH 03866, (US), (applicant designated states: DE;FR;GB;IT)
INVENTOR:
Spinney, Barry Alan, 22 Anthony Road, Wayland, MA 01778, (US)
LEGAL REPRESENTATIVE:
Betten & Resch (101031), Reichenbachstrasse 19, 80469 Munchen, (DE)

PATENT (CC, No, Kind, Date): EP 594196 A1 940427 (Basic)
EP 594196 B1 990331
APPLICATION (CC, No, Date): EP 93117155 931022;
PRIORITY (CC, No, Date): US 964738 921022
DESIGNATED STATES: DE; FR; GB; IT
INTERNATIONAL PATENT CLASS: H04L-012/46;
ABSTRACT WORD COUNT: 96

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9913	878
CLAIMS B	(German)	9913	861
CLAIMS B	(French)	9913	1056
SPEC B	(English)	9913	9177
Total word count - document A			0
Total word count - document B			11972
Total word count - documents A + B			11972

...SPECIFICATION of lookup records in this set. A set of lookup records is organized as a perfectly balanced, breadth-first binary **tree** . To obtain the required address, a binary search of this **tree** is done, and since the maximum depth of any **tree** is three, the maximum number of reads required is four - one to read the **pointer** from the hash **bucket** and the **tree** size, and three reads (maximum) to traverse a **tree** . A breadth-first storage representation is chosen because storage allocation for a breadth-first **tree** is never greater than the number of elements in the **tree** . For example, a hash **bucket** which **points** to five entries will take exactly five lookup records - with no lookup records required to be empty.

The second technique used in the combination is to handle the reasonably-rare case when more than seven 48-bit **addresses** hash to the same **bucket** . To handle this case, one of the addresses is simply put in a CAM **memory** chip that is present anyway. Also, it is noted that a destination address is compared to the CAM contents anyway...its higher ordered bits, indexed for one-of-22K selection of the word 91 in the table. The translation table **pointer** 93 returned by the hash **bucket** 90 in the hash table 89 is used to select a breadth-first balanced binary **tree** as illustrated in Figure 7. The **trees** are stored in a translation table 94 in **memory** 21, and each **tree** has between one and seven entries, as indicated by the size field 92. The binary **tree** cannot cross a block boundary in the translation table 94 in **memory** 21. The ordering of entries 96 in a breadth-first balanced binary **tree** for various table sizes is illustrated in Figure 7. Note that there are from one to seven entries (each entry...

...CLAIMS the step of, in parallel with said hashing, comparing said input address with a stored address in a content addressable **memory** (23).

3. A method according to claim 1 including the step of:
hashing a set of network addresses to generate...

...one entry at each index, and each entry contains a number of said hash buckets (90), wherein each of said **pointers** (93) in said hash **buckets** (90) identifies a binary **tree** of records in said translation (94) which can contain a number of said records as determined by said size value...

...of:
hashing set of network addresses to generate said hash table (89), and, if more than seven of said network **addresses** hash to a given one of said hash **buckets** (90), storing an overflow network **address** in a content addressable **memory** (23).

5. A method according to claim 4 including the step of, in parallel with said hashing, comparing said input address with any said stored overflow network address in said content addressable **memory** (23) including the step of
indexing into said translation table (94) with a value of said stored overflow network address...

...A method according to claim 1 including the step of storing said hash table (89) and said translation in a **memory** separate from a processor device performing said steps of hashing and comparing, including the step of sending said packet containing...

...comparator means for, in parallel with said hashing, comparing said input address with a stored address in a content addressable **memory** (23) including means for hashing a set of network addresses to generate said hash table (89), further including means for...

...one entry at each index, and each entry contains a number of said hash buckets (90) wherein each of said **pointers** (93) in said hash **buckets** (90) identifies a binary **tree** of records in said translation table (94) which can contain a number of said records as determined by said size...

...hashing a set of network addresses to generate said hash table (89), and, if more than seven of said network **addresses** hash to a given one of said hash **buckets** (90), storing an overflow network **address** in a content addressable **memory** (23), said apparatus furhter including means for, in parallel with said hashing, comparing (101) said input address with any said stored overflow network address in said content addressable **memory** (23).

10. Apparatus according to claim 7 including means for indexing into said translation table (94) with a value of...

...than by said pointer (93), said apparatus including means for storing said hash table (89) and said translation in a **memory** separate from a processor device performing said steps of hashing and comparing, said apparatus further including means for sending said...

17/3,K/11 (Item 11 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00359797
Bucket-oriented route planning method, and navigation system comprising a route planner for carrying out such a method.
Verfahren fur parzellenorientierte Streckenplanung sowie Navigationssystem mit einem Streckenplaner zur Durchfuhrung eines derartigen Verfahrens.
Procede pour prevoir un itineraire sur la base de parcelles ainsi que systeme de navigation muni d'un dispositif pour prevoir un itineraire et servant a mettre
PATENT ASSIGNEE:
Koninklijke Philips Electronics N.V., (200769), Groenewoudseweg 1, 5621 BA Eindhoven, (NL), (applicant designated states: CH;DE;ES;FR;GB;IT;LI;SE)
INVENTOR:
Verstraete, Rick Achiel, c/o INT. OCTROOIBUREAU B.V. Prof. Holstlaan 6,

NL-5656 AA Eindhoven, (NL)
LEGAL REPRESENTATIVE:
Strijland, Wilfred et al (21291), INTERNATIONAAL OCTROOIBUREAU B.V. Prof. Holstlaan 6, NL-5656 AA Eindhoven, (NL)
PATENT (CC, No, Kind, Date): EP 369539 A1 900523 (Basic)
EP 369539 B1 930519
APPLICATION (CC, No, Date): EP 89202867 891113;
PRIORITY (CC, No, Date): NL 882833 881117
DESIGNATED STATES: CH; DE; ES; FR; GB; IT; LI; SE
INTERNATIONAL PATENT CLASS: G01C-021/20; G06F-015/60;
ABSTRACT WORD COUNT: 59

LANGUAGE (Publication,Procedural,Application): English; English; Dutch
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	EPBBF1	739
CLAIMS B	(German)	EPBBF1	522
CLAIMS B	(French)	EPBBF1	658
SPEC B	(English)	EPBBF1	4365
Total word count - document A			0
Total word count - document B			6284
Total word count - documents A + B			6284

...SPECIFICATION digital data: one CD can store 4800 Mbit. This is more than thousand times the storage capacity of the largest **RAM** semiconductor **memory** available at present. The access time of a CD is much shorter than that of **a** magnetic **tape** cassette, but longer than that of a semiconductor **memory** and it is certainly not negligibly short.

The invention relates to a method of determining an optimum **route** between a starting position and a destination position on the basis of topographical and traffic information by repeated selection of vectors and expansion of a search **tree** which contains vectors which form already planned sub-routes, to each vector there being assigned a weighting **factor** and for each sub-route there being determined a cumulative weighting factor by addition of the weighting factors of the ...

...the already planned sub-route.

The invention also relates to a navigation system comprising a route planner which includes:

- a **memory** for the bucket-wise storage of topographical and traffic **information** ;
- an input/output unit for the input **and** output of **information** concerning starting position and destination position;
- **a** processor which is programmed so that, via repeated selection of vectors and expansion of a search **tree** containing vectors which form already planned sub-routes, an optimum route is calculated from a given starting **position** to a given destination position on the basis of weighting factors assigned to each vector.

A method of this kind...

...a working memory to be used in conjunction with the method and the navigation system.

In the article "CAR Guide- **on** -board computer for automobile route guidance", M. Sugie et al., AFIPS Conference Proceedings, 1984 National Computer Conference, Las Vegas, Nevada...

...information is stored and also comprises a working memory whereto there are transferred from the background memory, under the control **of** the processor, only the **buckets** **which** have been selected on the basis of an evaluation value which is determined by a sum of the weighting factors...

...buckets in the working memory being used by the processor for the repeated selection of vectors and the expansion of **the** search **tree** .

It is thus achieved that the instantaneously required **information** is always present in the working **memory** .

BRIEF DESCRIPTION OF THE FIGURES

information is stored and also comprises a working **memory** whereto there are transferred from the background **memory** only the buckets which have been selected on the basis of an evaluation value which is determined by a sum...

...an estimated fictitious sub-route yet to be followed via the relevant bucket, only vectors from buckets in the working **memory** being used for the repeated selection of vectors and the expansion of the search **tree** .

3. A method as claimed in Claim 1, characterized in that said evaluation value for a bucket in the working **memory** is determined by the most attractive evaluation value of all vectors in the relevant bucket which are listed on a candidate list, and is determined for a neighbouring bucket of a bucket in the working **memory** by the addition of the weighting factor of an idealised straight path between a current position and a point z, being a **point** of intersection of a connecting line between centres of said **bucket** and said neighbouring bucket and a common boundary (or the prolongation thereof) of said bucket and said neighbouring bucket, the...

17/3,K/21 (Item 10 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00451467 **Image available**
METHOD FOR IMPLEMENTING AN ASSOCIATIVE MEMORY BASED ON A DIGITAL TRIE STRUCTURE
PROCEDE DE MISE EN OEUVRE D'UNE MEMOIRE ASSOCIATIVE SUR LA BASE D'UNE ARBORESCENCE NUMERIQUE
Patent Applicant/Assignee:
NOKIA TELECOMMUNICATIONS OY,
TIKKANEN Matti,
IIVONEN Jukka-Pekka,
Inventor(s):
TIKKANEN Matti,
IIVONEN Jukka-Pekka,
Patent and Priority Information (Country, Number, Date):
Patent: WO 9841931 A1 **19980924**
Application: WO 98FI190 19980304 (PCT/WO FI9800190)
Priority Application: FI 971065 19970314
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM
GW HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX
NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US UZ VN YU ZW GH
GM KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH DE DK ES FI
FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD TG
Publication Language: English
Fulltext Word Count: 10126

Patent and Priority Information (Country, Number, Date):
Patent: ... **19980924**
Fulltext Availability:
Detailed Description
Claims

English Abstract
The invention relates to a method for implementing a **memory** . The memory is implemented as a directory structure comprising a **tree** -shaped **hierarchy** having nodes at several different levels, wherein an individual node can be (i) a **trie** node comprising an array wherein an individual element may contain the address of a lower node in the **tree** -shaped **hierarchy** and wherein an individual element may also be empty, the number of elements in the array corresponding to a power of two, or (ii) a **bucket** containing at least one element so that the type of an

wherein when condition (c) is...

...h a r a c t e r i z e d in that in
at least part of the **directory** structure, sets of successive **trie** nodes are replaced with compressed nodes in such a way that an individual set made up by successive trie nodes...

...a r a c t e r i z e d in that
replacement is carried out in the entire **directory** structure so that all said sets are replaced with compressed nodes.
1 5 4. A method as claimed in claim...claim 3, c h a r a c t e r i z e d in that in
all uncompressed **trie** nodes of the **memory** , at least two addresses to a lower
level node are maintained.

9 A method as claimed in claim 2, c h a r a c t e r i z e d in that the **directory** structure stores for each node information on the fact whether an uncompressed **trie** node, a compressed **trie** node or a **bucket** is concerned.
1 0. A method for implementing a **memory** , in which **memory** data is stored as data units for each of which a dedicated storage space is assigned
in the **memory** , in accordance with which method
1 0 - the **memory** is implemented as a **directory** structure comprising a **tree** -shaped **hierarchy** having nodes at several different levels, wherein an individual node can be (i) an internal node comprising an array wherein an individual element may contain the address of a lower node in the **tree** -shaped **hierarchy** and wherein an individual element may also be empty, the number 1 5 of elements in the array corresponding to...

...of which is one from a group including a pointer to a stored data unit and a pointer to another **directory** structure, - address computation performed in the directory structure com
prises the steps of
- (a) selecting in the node at the...

17/3,K/24 (Item 13 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00283453 **Image available**
PREDICTIVE DISK CACHE SYSTEM
SYSTEME D'ANTEMEMOIRE A DISQUE PREDICTIF
Patent Applicant/Assignee:
OAKLEIGH SYSTEMS INC,
Inventor(s):
DORNIER Pascal,
Patent and Priority Information (Country, Number, Date):
Patent: WO 9501600 A1 **19950112**
Application: WO 94US7882 19940701 (PCT/WO US9407882)
Priority Application: US 9386722 19930702
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
CN JP AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE
Publication Language: English
Fulltext Word Count: 4847

Patent and Priority Information (Country, Number, Date):
Patent: ... **19950112**
Fulltext Availability:
Detailed Description
Publication Year: **1995**

Detailed Description

... to change
hardware or software addresses as well as to customize
overall cache algorithms, such as: cache size, application
shared **memory** allocation ratios; partitioning of the set
associations, LAN network nodal assignments and
priorities, and system wide defaults. Options include ...Along with
them, an associated chart gives the user a
past performance 11cache hit" ratio for each of the
previously **tried** sequence **table** routines. Set-up also
establishes batch files inside command routines at "boot
.up", as well as establishing sub- **directories** needed for
the Predictive Cache System. An optional utility
benchmark program displays current performance data
related to cache hits, hit...

...step 16, so it
can manipulate exiting disk caching software batch
routines. This assures optimization by not allocating
valuable system **memory** to two or three caches (three in
the case of both an existing hardware and software cache),
and also saves...

```
File 275:Gale Group Computer DB(TM) 1983-2004/Sep 20
         (c) 2004 The Gale Group
File 621:Gale Group New Prod.Annou.(R) 1985-2004/Sep 20
         (c) 2004 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2004/Sep 20
         (c) 2004 The Gale Group
File  16:Gale Group PROMT(R) 1990-2004/Sep 20
         (c) 2004 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 148:Gale Group Trade & Industry DB 1976-2004/Sep 20
         (c)2004 The Gale Group
File 624:McGraw-Hill Publications 1985-2004/Sep 17
         (c) 2004 McGraw-Hill Co. Inc
File  15:ABI/Inform(R) 1971-2004/Sep 18
         (c) 2004 ProQuest Info&Learning
File 647:CMP  Computer Fulltext 1988-2004/Sep W2
         (c) 2004 CMP Media, LLC
File 674:Computer News Fulltext 1989-2004/Aug W4
         (c) 2004 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2004/Sep 20
         (c) 2004 The Dialog Corp.
File 369:New Scientist 1994-2004/Sep W1
         (c) 2004 Reed Business Information Ltd.

Set     Items   Description
S1    1512693   MEMOR??? OR RAM OR DRAM OR SRAM OR SDRAM OR RDRAM OR SLDRAM
              OR SGRAM OR DRDRAM OR ROM OR PROM OR EPROM OR EEPROM OR FPO -
             OR EDO
S2     902957   DIRECTORY OR DIRECTORIES OR HIERARCH? OR TREE? ?
S3          3   TRIE()NODE? ?
S4    1724006   TABLE? ? OR LUT? ?
S5          8   (POINT??? OR ADDRESS???)(5N)((LOWER OR DEEPER)(3N)NODE? ?)
S6       2098   BUCKET? ?(10N)(DATA OR INFORMATION OR POINT??? OR ADDRESS?-
              ?? OR S2)
S7          1   S1(50N)S2(50N)TRIE? ?(50N)BUCKET? ?
S8         12   S1(50N)S2(50N)S6
S9          0   TRIE? ?(20N)S5
S10        23   S3 OR S5 OR S7:S8
S11        20   RD (unique items)
```

11/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

02806493 SUPPLIER NUMBER: 118596883 (USE FORMAT 7 OR 9 FOR FULL TEXT
)
Handling memory fragmentation: fragmentation can be a sticky problem. How memory allocation occurs determines whether, when, and how memory fragmentation becomes an issue.(design feature)
Lindblad, Jan
EDN, 49, 12, 77(4)
June 10, 2004
ISSN: 0012-7515 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 3204 LINE COUNT: 00276

... other allocators described in this article, don't carve out new blocks as needed from the beginning of the managed **memory** . The defining commonality is that blocks are split and joined, but not arbitrarily. Each block has a friend, or "buddy...

...Buddy allocators store blocks in data structures more advanced than linked lists. Often, the structures are combinations or variations of **buckets** , **trees** , and heaps. It is hard to describe in general how buddy allocators work, because the technique varies with the selected...
...write, and their properties may vary. Usually, they limit fragmentation to some degree.

Fixed-size allocators are somewhat like first- **tree** algorithms. There is usually more than one free list, and, most important, all blocks in the same free list are identical in size. There are at least four pointers: MSTART points to the beginning of the managed **memory** , MEND points to the end of the managed **memory** , MBREAK points to the end of the used **memory** between MSTART and MEND, and PFREE(n) is an array of pointers to any free **memory** blocks. In the beginning, PFREE(*) is NULL, and MBREAK points at MSTART. When an allocation request comes in, the system...

11/3,K/2 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

02609741 SUPPLIER NUMBER: 87012021 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Burst tries: a fast, efficient data structure for string keys.(Statistical Data Included)
Heinz, Steffen; Zobel, Justin; Williams, Hugh E.
ACM Transactions on Information Systems, 20, 2, 192(32)
April, 2002
DOCUMENT TYPE: Statistical Data Included ISSN: 1046-8188
LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 14033 LINE COUNT: 01113

... of a single container. When a container is deemed to be inefficient, it is burst, that is, replaced by a **trie node** and a set of child containers that, between them, partition the original container's strings. Thus there are two major...respectively, follows) c. Thus a set of TST nodes connected by left and right pointers are a representation of a **trie node** . These can be rebalanced on access. The central pointer is for strings starting with c, thus corresponding to the `c' pointer of a **trie node** . TSTs are slower than tries, but more compact. We report experiments with compact tries and TSTs below. Tries are discussed...of pointers, each of which may point to either a **trie node** or a container, and a single empty-string pointer to a record. The...
...is the root of the access trie and the current depth i is 1.

While the current object is a **trie node** t of depth i (less than or equal to) n,

(...c.sub.i)th element of t's array p, and

(b) Increment i.

If the current object is a **trie node** t of depth i = n + 1, the current object becomes the object pointed to by the empty-string pointer,

...children as subtrees (a list trie) (de la Briandais 1959; Knuth 1968). A list implementation offers space savings if a **trie node** has only a few children, when an array of fixed size would consist largely of null pointers. However, the space...

...on theoretical analysis of a list trie, Sussenguth has suggested that the expected search time can be minimised when list **trie nodes** stop branching when there are less than six keys. Instead of a further branching, the keys should be kept directly...

...analysis concerns a slow variant of tries and cannot be applied to array tries.

Another option is to implement a **trie node** as a binary search tree. Bentley and Sedgewick (1997) propose the TST, where the **trie nodes** are binary search trees. Clement et al. (2001) analyse and empirically test these hybrid trie structures. Array tries, list tries...

...large set of keys.

The second group of proposals to reduce the size of tries deal with the number of **trie nodes** . In a standard trie, all characters of all strings are represented by pointers between **trie nodes** . However, in natural-language applications the **trie nodes** near the leaf levels tend to be sparse. (This sparsity does not arise, for example, with genomic data.) A simple way to reduce the number of **trie nodes** is to omit chains of nodes that have only a single descendant and lead to a leaf. We refer to...

pointers between nodes

...above--an alternative implementation known as a compact Patricia trie. Both implementations lead to a decrease in the number of **trie nodes** , a saving that is partly offset by a more complex structure and more complex traversals.

The value of the Patricia...for TREC1, only 15 of 88,016 nodes are single-descendant nodes, while for Web M around 2% of access **trie nodes** are single-descendant nodes.

In many descriptions of Patricia tries, each node has only two pointers, and the next bit...

...one node occupy the same location as full pointers in another. Purdin (1990) proposed reduction of the size of a **trie node** by using a compressed bitmap in each node to indicate which children slots are used. Because of the overhead of...

...a good compromise between memory usage and low access costs in a trie. The idea is to collapse subtrees of **trie nodes** completely and store the (suffix) strings represented by the subtree in a BST that is used as a leaf node...

...optimized trie. One heuristic is designed to meet the objective that the resulting trie has only a minimal number of **trie nodes** but the worst-case binary search time does not exceed a fixed quantity. Another minimizes the worst-case binary search time for a given maximal number of **trie nodes** . An additional heuristic tries to solve both objectives. Although those heuristics cannot be applied to dynamic sets of strings, the ...the cost of searching the container is offset by the saving of not having to traverse a large number of **trie nodes** . Together, these savings allow in-memory processing of larger sets of strings than was previously possible, in much less time...

11/3,K/3 (Item 3 from file: 275)

File 275:Gale Group Computer DB(TM)
The Gale Group. All rts. reserv.

SUPPLIER NUMBER: 16288888 (USE FORMAT 7 OR 9 FOR FULL TEXT)
ous network convergence. (Net Worth) (Column)
ven
, v12, n12, p21(6)

Nov, 1994
DOCUMENT TYPE: Column ISSN: 0742-3136 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2403 LINE COUNT: 00203

... of configuring and maintaining two separate networks. Most PC NFS client software for DOS/Windows costs $200 to $400 per **node** . Even at the **lower** price **point** , a 50-user network employing PC NFS would cost $10,000 for the basic TCP/IP client software. Compare this...

11/3,K/4 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01556401 SUPPLIER NUMBER: 14330746 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Options for change: disk fragmentation is one of the many trials faced by the VAX systems manager. (survey of systems management software) (Buyers Guide)
Sethi, Joginder
DEC User, p39(5)
Sept, 1992
DOCUMENT TYPE: Buyers Guide ISSN: 0263-6530 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1559 LINE COUNT: 00123

... the MONITOR FCP utility to examine the window turn rate.
You may also need to increase the size of the **directory** data cache if the **directory** files contain more than 500 entries, or if files are frequently added and deleted from **directories** . The performance of the **directory** file can be compared directly with an RMS indexed file which has been subject to continuous additon and deletion of records without a file reorganisation. However, do not waste **memory** by allocating too large a value, and always make the changes in co-ordination with Autogen.
OPTION 10
Buy a file optimiser/disk defragmentation package (see survey).
OPTION 11
Introduce RMS buffering. RMS parameters such as BUFFER COUNT, INDEX FILL, **DATA** FILL, ALLOCATION, GLOBAL BUFFERS, and **BUCKET** SIZE can improve the performance of your applications. Some of the RMS features help reduce the impact of fragmentation, but...

11/3,K/5 (Item 5 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01537087 SUPPLIER NUMBER: 12042188 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Lexical analysis using search tries. (Tutorial)
Stevens, John W.M.
C Users Journal, v10, n4, p67(18)
April, 1992
DOCUMENT TYPE: Tutorial ISSN: 0898-9788 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1663 LINE COUNT: 00124

... type, the trie search function is called. The function TrieSrch() begins by attempting to find the input character in the **trie node** . TrieSrch() accepts a pointer to a node of a trie, a character to search for, and a pointer to a...

...for storing the word read from the input file. The function uses a binary search because the characters in a **trie node** are stored in sorted order.
If the input character is found in the **trie node** , TrieSrch() saves it in the word buffer. If the matching character in the **trie node** has a pointer to a child **trie node** , TrieSrch() reads another character from the file and calls itself recursively. If the return value from the

recursive call indicates...

...the input character to this call is returned.
If the matching character does not have a pointer to a child **trie node** , the keyword buffer is NUL-terminated and the token value stored with the matching character is returned. If the input character is not found in the **trie node** , TrieSrch() NUL-terminates the keyword buffer and returns a value indicating that the character was not found.
Figure 2 presents...

11/3,K/6 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01534809 SUPPLIER NUMBER: 12545976 (USE FORMAT 7 OR 9 FOR FULL TEXT)
UTP for flexibility. (unshielded twisted pair cabling for networks) (Courtaulds Advanced Materials)
IBM System User, v13, n6, p17(1)
June, 1992
ISSN: 0950-303X LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1028 LINE COUNT: 00077

...ABSTRACT: corporate center in London. The disadvantages of UTP cabling include the need for short cable runs, possible interference, and the **lower** number of **nodes** supported and were all **addressed** by the company in the initial planning stages. It installed a Token-Ring network because of employee experience in the...

11/3,K/7 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01532053 SUPPLIER NUMBER: 12585843 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Zortech: Symantec Corp. (Software Review) (Product Wrap-Up)(one of six class libraries evaluated) (Evaluation)
O'Brien, Larry
Computer Language, v9, n10, p84(1)
Oct, 1992
DOCUMENT TYPE: Evaluation ISSN: 0749-2839 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 183 LINE COUNT: 00014

ABSTRACT: Symantec Corp's Zortech compiler includes class libraries that emphasize encapsulating areas of DOS and **memory** management. This emphasis is odd considering that Zortech is one of the most portable native compilers for C++. The most frequently used in- **memory** collection classes are provided, including dynamic arrays, singly and doubly linked lists, binary **trees** and hash tables; also included are **buckets** and virtual arrays for managing file-based **data** . A binary-coded decimal class is supplemented with a class specialized for manipulating money figures and a class for handling extended two's complement integers. The interrupt vector and DOS critical error classes are very useful, but the **Directory** and Filename DOS classes are nothing special. A series of classes for implementing a text-based windowing interface is also...
Zortech includes the most commonly used in- **memory** collection classes (dynamic arrays, singly and doubly linked lists, binary **trees** , and hash tables) as well as **buckets** and virtual arrays for managing file-based **data** . Zortech has a BCD class supplemented with a class specialized for manipulating money figures and a class that handles extended two's complement integers.
The DOS classes for **Directory** and Filename are useful but not earthshaking (I'm normally not the sort to say "Oh, I could reproduce that ...

11/3,K/8 (Item 8 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)

01511086 SUPPLIER NUMBER: 11744058 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Practical dictionary management for hardware data compression. (for the second Ziv-Lempel data compression scheme) (Technical)
Bunton, Suzanne; Borriello, Gaetano
Communications of the ACM, v35, n1, p95(11)
Jan, 1992
DOCUMENT TYPE: Technical ISSN: 0001-0782 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 5779 LINE COUNT: 00466

... performance figures illustrating the merit of LRU, see [5]. Typically, an LRU implementation uses a linked list of pointers to **trie nodes** to order nodes by recency of use. The list must be doubly linked to allow constant time deletions. For constant time search, each **trie node** also keeps a pointer to its associated link in the LRU queue. Since four extra pointers per dictionary entry are...the root receives a new tag that is one larger than its old tag, modulo the maximum number of transient **trie nodes** , N. With a full dictionary, the root's tag will equal the tag of the leaf node that terminates the...routine communicates with the compressor by sharing two variables, free and node which are the addresses of the next unoccupied **trie node** and the currently visited **trie node** , respectively. A third variable, prev-node can be shared, or the tagging routine can keep its own copy.
Until the...

...in a variable input rate, but bounded probe hashing lessens the resulting buffering burden at the cost of occasional "lost" **trie nodes** . These node losses do not effect the correctness or appreciably degrade the performance of TAG or any other scheme presented...

...can write and match simultaneously for storing the dictionary
Tag, Tag': two 1K X 10-bit RAMs for storing the **trie node** -to-tag mapping and its inverse
control: a data path composed of the following:
* 90 bits of state (includes a...

11/3,K/9 (Item 9 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01377233 SUPPLIER NUMBER: 09475659 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Sharing the task of on-line transaction processing. (tutorial)
Beach, Paul
DEC User, p35(2)
August, 1990
DOCUMENT TYPE: tutorial ISSN: 0263-6530 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1957 LINE COUNT: 00155

... time, should a number of users 'commit' together. Such an implementation implies some sort of page or row versioning in **memory** , in order to ensure that each user sees a consistent picture of the database. Otherwise, when commit is invoked, updated...

...a row of data.
The access of a hash bucket does not involve having to traverse a number of B- **tree** index nodes to access the relevant leaf. instead, hash bucket access is usually determined by an algorithm that points to...to access hashed data, provided that data can be held on the same page of the hash bucket. A B- **tree** access to data could be a number of I/Os, depending on the depth of the B- **tree** .
Although hash **buckets** reduce potential 1/0, it is the area of insert and update in which they really become important. The lowest level of B- **tree** index structure locking tends to be node level; the lowest

level for hash structures is usually at **bucket** level. Since a B- **tree** node could hold pointers to a number of rows, all rows referenced by the B- **tree** node are effectively locked should a pointer need updating or inserting. The hash **bucket** usually contains just one pointer to one row, so other users are unlikely to be affected.

In addition, hash **buckets** do not need to be rebalanced. Should a B- **tree** node need to split, and the structure need to be rebalanced, whole tables may effectively be locked while the rebalancing is taking place. Some database vendors who do not support hash keys have **tried** to implement more sophisticated B- **tree** structures. However, it is important to recognise that B- **tree** index structures are inherently inefficient where OLTP applications are concerned. OLTP systems lend themselves to demand page buffering rather than...

...Because users are only likely to access specific pages for data, it is generally unnecessary to bring unwanted pages into **memory** . Therefore, a large number of small buffers are usually the most efficient. At worst, a database designer is able to...
...to define the number of buffers required to satisfy the number of users that table is supporting, without wasting any **memory** .

The facility to place and cluster data in an OLTP database is particularly useful. Because transactions are predetermined, the database ...

11/3,K/10 (Item 10 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01289566 SUPPLIER NUMBER: 07125346 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Tech notebook; a forum for sharing solutions to technical problems. (column)
Mirecki, Ted
PC Tech Journal, v7, n4, p117(3)
April, 1989
DOCUMENT TYPE: column ISSN: 0738-0194 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2418 LINE COUNT: 00188

... for itself the two highest-numbered page frames outside of the primary frames. Note that besides reserving pages of expanded **memory** (as does any well-behaved program that uses EMS), DOS claims exclusive use of page frames at specific addresses in the physical address space. No provision is made for this in the Expanded **Memory** Specification.

Furthermore, DOS hides these frames from all other processes. Because DOS assumes that no other process can access its...

...mapping by switching to an alternate register set.

Consider the case where DOS caches a file allocation table (FAT) or **directory** in a buffer in one of its private page frames, then writes that buffer out to disk without ensuring that...

...still mapped to the frame. If the operating environment has switched the mapping context in the meantime, the FAT or **directory** gets overwritten with garbage, and some or all of the **data** on the disk goes into the bit **bucket** .

A more likely, but fortunately less damaging failure occurs if DOS reserves page frames within conventional **memory** . Because DOS takes the highest-numbered pages, and the EMM assigns higher numbers to frames at higher addresses, these pages are invariably at the top of conventional **memory** , where DOS keeps the transient portion of COMMAND.COM. Using the same area for buffers, however, overwrites COMMAND.COM. This might not seem to be a major problem, because the low- **memory** portion of DOS checks the upper portion and reloads COMMAND.COM if it gets trashed; however, the check is performed...

11/3,K/11 (Item 11 from file: 275)

DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01248345 SUPPLIER NUMBER: 06990727 (USE FORMAT 7 OR 9 FOR FULL TEXT)
SST: The Seek Stopper. (Software Review) (One of 13 hard disk utility programs evaluated in 'Boosting Performance with a Well-Ordered Disk') (evaluation)
Mendelson, Edward
PC Magazine, v7, n17, p203(2)
Oct 11, 1988
DOCUMENT TYPE: evaluation ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 614 LINE COUNT: 00042

... if you use brute force to stop it by pressing Ctrl-Alt-Del or by turning off the machine, your **data** will disappear into the great bit-**bucket** in the sky. Most other disk packers guard against accidental **data** loss by keeping a spare copy of a cluster on-disk while transferring another copy from one part of the disk to another. SST does all its work in **memory** and doesn't update the file allocation tables or **directories** until it's finished.
You can run SST in a test mode that will run through its full procedures without...

11/3,K/12 (Item 12 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01192092 SUPPLIER NUMBER: 06048898
A new method for fast data searches with keys.
Litwin, Witold; Lomet, David B.
IEEE Software, v4, n2, p16(9)
March, 1987
ISSN: 0740-7459 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

...ABSTRACT: support key associative access to data utilizing key associative access to data utilizing key associative access to data utilizing both **tree** -indexing and hashing. Indexing allows for random anD sequential access to **data** . Hashing accesses large, multi- **bucket** nodes of **data** , reducing the index size for a particular file size and providing for a main **memory** index. Bounded disorder files feature an improved i-ratio, or file size to index size ratio, and their performance in...

11/3,K/13 (Item 1 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2004 The Gale Group. All rts. reserv.

01213903 Supplier Number: 43693313 (USE FORMAT 7 FOR FULLTEXT)
Bristol Babcock Announces Expanded Network Addressing in NETWORK 3000
News Release, p1
March 5, 1993
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 182

... operate as "pass-through" nodes, i.e. they can transferinofoTmation between levels. Each communication port on each node can directly **address** up to 127 **nodes** at the next **lower** level.

Although the theoretical limit for an entire network is 32,767 nodes, the local 127 node limit had been...

11/3,K/14 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

09830701 SUPPLIER NUMBER: 17778156 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Fire, floods and drought keep managers on watch. (real estate managers)
Weiss, Lois
Real Estate Weekly, v42, n4, p1(2)
August 30, 1995
LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 1005 LINE COUNT: 00080

... certain properties they have voluntarily cut down on the watering from three to two times a week and are ensuring **trees** are watered with **buckets** , as those are expensive to replace and take many years to fill out.
"We are using soaker hoses as much...

...the grounds like a tinderbox," he said. "We are working with our landscapers to maintain everything."
Jeffrey C. Gold, CPM, **RAM** , vice president of Marvin Gold Management that manages about 25,000 apartments in the boroughs, New Jersey and Long Island...

11/3,K/15 (Item 2 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

06771704 SUPPLIER NUMBER: 14795339 (USE FORMAT 7 OR 9 FOR FULL TEXT)
The development of indexing technology.
Chang, Roy
Library Software Review, v12, n3, p30(6)
Fall, 1993
ISSN: 0742-5759 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 5151 LINE COUNT: 00371

... first entry into it. Each node on a binary tree has a key, and it can, at most, have two **pointers** : one links to the **lower** -left **node** , and the other one links to the lower-right node. The node can be represented as: (p1 key p2), in...

11/3,K/16 (Item 3 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

05924662 SUPPLIER NUMBER: 12713107 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Cox moves more aggressively on fiber optics. (Cox Cable of San Diego Inc.)
Dawson, Fred
Multichannel News, v13, n25, p29(2)
June 22, 1992
ISSN: 0276-8593 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 1156 LINE COUNT: 00086

... will require higher power transmitters than are presently being installed or regeneration of the signal at the present node termination **point** for transmission to the **deeper nodes** .
S-A is leaning toward use of high-power transmitters with output split at the present node site over four...

11/3,K/17 (Item 4 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

05915789 SUPPLIER NUMBER: 12473603 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Will 1 GHz be enough bandwidth? (for cable television)
Dawson, Fred
Multichannel News, v13, n20, p41(1)
May 18, 1992

ISSN: 0276-8593 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 1412 LINE COUNT: 00109

... the firm is looking at using optical amplification to boosts a signal in order to extend the FSA system to **deeper node points** , perhaps employing a four-way splitter at the existing node to achieve a reach to four subnodes.
Right now, Fellows...

11/3,K/18 (Item 1 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

00690593 93-39814
Performance analysis of a main memory multi-directory hashing technique
Analyti, Anastasia; Pramanik, Sakti
Information Processing Letters v45n4 PP: 191-197 Mar 22, 1993
ISSN: 0020-0190 JRNL CODE: IPL

ABSTRACT: Hashing is a well-known technique in database systems that permits fast access to both disk-based and main **memory** -based databases. Hashing schemes for disk-based databases have been designed with the assumption that data reside on the disk during transaction processing. However, substantial performance gains can be achieved when data reside in main **memory** . The rapidly decreasing cost of random access **memory** (**RAM**) makes main **memory** databases a cost-effective solution to high-performance data management. A main **memory** multi- **directory** hashing technique, called Extendible Root Multi- **Directory** Hashing (ERMH), is presented and analyzed. ERMH is a dynamic hashing technique that yields optimal search in main **memory** databases. ERMH uses a **tree** -structured **directory** of height one. The number of index accesses to locate a record is 2. Optimal search can be obtained by extendible hashing with **bucket** size one because in this case **directory** entries **point** to at most one data record. ...

11/3,K/19 (Item 1 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext

01146244 CMP ACCESSION NUMBER: HPC19971201S0087
Let's pretend we just never saw this one (Kid Raves-Our Young Experts Rate Software)
Carol Ellison
HOME PC, 1997, n 412, PG232
PUBLICATION DATE: 971201
JOURNAL CODE: HPC LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: Reviews
WORD COUNT: 396

... I can't get them colored in-it's too hard," said John, 4, who had trouble positioning the paint **bucket** accurately to spill color into the leaves on a **tree** .
John and Ryan, also 4, picked up one of the activity books that come with each disc and busied themselves...

...game.
"No fun," she said with a shrug. "Nothing to do."
$19.95 per title (Windows 3.1/95 CD- **ROM** , Macintosh CD- **ROM**) from Mind Magic, (800) 762-6443, (941) 355-3057, www.mindmagic.com Circle # 521

11/3,K/20 (Item 2 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext

00559902 CMP ACCESSION NUMBER: EET19900205S3473
Software manages C data structures
RAY WEISS
ELECTRONIC ENGINEERING TIMES, 1990, n 576, 39
PUBLICATION DATE: 900205
JOURNAL CODE: EET LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: DES
WORD COUNT: 1392

... ZZssEnd (loop to retrieve next item in reverse order), ZZssBackward (get next-tolast item, for double-linked organizations) and ZZssSort. **Tree** operations include ZZssOrg, ZZssSingle/ Double- **Tree** , ZZssAdd, ZZss Append (add new sibling), ZZssDelete-Forward, ZZssParent, ZZssChild, ZZssSet (set first child), ZZssInsert, ZZssBackward, ZZssAssRetrace and ZZssAssTraverse.
OrgC...

...mechanisms to save individual objects or full organizational sets. Those data structures can be saved in binary form to conserve **memory** or in ASCII for portability to other, heterogeneous systems.
Organizations can also be deleted on a large scale with a...

...ZZssClear command.
One nice feature is OrgC's provision of hash table operations. Hash tables are an indirect method to **address** a set of items or **buckets** . Instead of searching the set directly, a function maps some input key into an index, which points directly to the...

File 347:JAPIO Nov 1976-2004/May(Updated 040903)
(c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200459
(c) 2004 Thomson Derwent
File 348:EUROPEAN PATENTS 1978-2004/Sep W02
(c) 2004 European Patent Office
File 349:PCT FULLTEXT 1979-2002/UB=20040916,UT=20040909
(c) 2004 WIPO/Univentio

Set	Items	Description
S1	34	AU=(IIVONEN J? OR TIKKANEN M?)
S2	15	S1 AND TRIE? ?
S3	6	S2 AND BUCKET? ?

3/5/1 (Item 1 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00762399 **Image available**
MEMORY BASED ON A DIGITAL TRIE STRUCTURE
MEMOIRE BASEE SUR UNE STRUCTURE D'ARBRE DERIVE NUMERIQUE

Patent Applicant/Assignee:
NOKIA NETWORKS OY, Keilalahdentie 4, FIN-02150 Espoo, FI, FI (Residence), FI (Nationality), (For all designated states except: US)
Patent Applicant/Inventor:
TIKKANEN Matti , Jahtimestarintie 18, FIN-02940 Espoo, FI, FI (Residence), FI (Nationality), (Designated only for: US)
IIVONEN Jukka-Pekka , Franzeninkatu 3 B 58, FIN-00500 Helsinki, FI, FI (Residence), FI (Nationality), (Designated only for: US
Legal Representative:
PATENT AGENCY COMPATENT LTD, Pitkansillanranta 3 B, FIN-00530 Helsinki, FI

Patent and Priority Information (Country, Number, Date):
Patent: WO 200075805 A1 20001214 (WO 0075805)
Application: WO 2000FI381 20000428 (PCT/WO FI0000381)
Priority Application: FI 991262 19990602
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AE AG AL AM AT AU AZ BA BB BG BR BY CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG US UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Main International Patent Class: G06F-017/30
Publication Language: English
Filing Language: Finnish
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 7558

English Abstract
The invention relates to a method for implementing a memory and to a memory arrangement. The memory is implemented as a directory structure comprising a tree-shaped hierarchy having nodes at several different hierarchy levels. The directory structure employs width-compressed nodes in which non-nil pointers are physically stored and additionally a bit pattern which has one bit for each element and wherein e.g. a 1-bit indicates that the content of the element corresponds to a pointer pointing downward in the directory structure. The physical storage location in the node, corresponding to the element table index associated with the search word, is determined on the basis of the bit pattern. To enable rapid determination that requires little memory capacity, a search table is stored in which numbers of bits having value 1 are stored in different combinations of a word whose total number of bits is a predetermined portion, preferably half, of the number of bits in said bit pattern. In the bit pattern, all bits corresponding to an index that is greater than the element table index are changed to zero, and a given number of reading operations are carried out from the search table in accordance with the value of the element table index as compared to the total number of bits in said word, the numbers obtained being added together when there are more than one reading operations.

French Abstract
L'invention concerne un procede de mise en oeuvre d'une memoire ainsi qu'un agencement de memoire. La memoire est mise en oeuvre sous la forme d'une structure d'annuaire comprenant une hierarchie arborescente presentant des noeuds en plusieurs niveaux hierarchiques differents. La

structure d'annuaire utilise des noeuds compresses en largeur dans lesquels des pointeurs non nuls sont stockes physiquement et en plus une configuration binaire ayant un bit pour chaque element et dans laquelle, par exemple, un bit 1 indique que le contenu de l'element correspond a un pointeur pointant vers le bas dans la structure d'annuaire. L'emplacement de stockage physique dans le noeud, correspondant a l'index de la table d'elements associe a chaque mot, est determine sur la base de la configuration binaire. Pour permettre une determination rapide necessitant peu de capacite memoire, une table de recherche est stockee dans laquelle des nombres de bits ayant une valeur 1 sont stockes dans differentes combinaisons d'un mot dont le nombre total de bits est une partie predeterminee, de preference la moitie, du nombre des bits se trouvant dans ladite configuration binaire. Dans la configuration binaire, tous les bits correspondant a un index superieur a l'index de la table d'elements sont changes en zero, et un nombre donne d'operations de lecture est execute dans la table de recherche selon la valeur de l'index de la table d'elements compare au nombre total de bits se trouvant dans le mot, les nombres obtenus etant additionnes lorsqu'il y a plus d'une operation de lecture.

Legal Status (Type, Date, Text)
Publication 20001214 A1 With international search report.
Examination 20010208 Request for preliminary examination prior to end of 19th month from priority date

3/5/2 (Item 2 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00762398 **Image available**
FUNCTIONAL MEMORY BASED ON A TRIE STRUCTURE
MEMOIRE FONCTIONNELLE A STRUCTURE DE RECHERCHE
Patent Applicant/Assignee:
NOKIA NETWORKS OY, Keilalahdentie 4, FIN-02150 Espoo, FI, FI (Residence), FI (Nationality), (For all designated states except: US)
Patent Applicant/Inventor:
TIKKANEN Matti , Jahtimestarintie 18, FIN-02940 Espoo, FI, FI (Residence), FI (Nationality), (Designated only for: US)
IIVONEN Jukka-Pekka , Franzeninkatu 3 B 58, FIN-00500 Helsinki, FI, FI (Residence), FI (Nationality), (Designated only for: US
Legal Representative:
PATENT AGENCY COMPATENT LTD, Pitkansillanranta 3 B, FIN-00530 Helsinki, FI
Patent and Priority Information (Country, Number, Date):
Patent: WO 200075804 A1 20001214 (WO 0075804)
Application: WO 2000FI380 20000428 (PCT/WO FI0000380)
Priority Application: FI 991261 19990602
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AE AG AL AM AT AU AZ BA BB BG BR BY CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG US UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Main International Patent Class: G06F-017/30
Publication Language: English
Filing Language: Finnish
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 7383

English Abstract

The invention relates to a method for implementing a functional memory and to a memory arrangement. The memory is implemented as a directory structure comprising a tree-shaped hierarchy having nodes at several different hierarchy levels. In the directory structure, pointers are first added to nodes whose table contains a given first number of elements and which are width-compressed nodes. To maximize the performance of the functional **trie** structure, addition of a pointer to an individual width-compressed node is permitted until the number of pointers in the node corresponds to a given predetermined threshold value that is smaller than said first number. The width-compressed node is converted to a cluster of nodes made up by a parent node (N50) and separate child nodes (N51...N54) as soon as the number of pointers to be accommodated in the width-compressed node exceeds said threshold value.

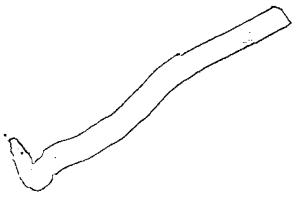
French Abstract
La presente invention concerne un procede de mise en oeuvre d'une memoire fonctionnelle et une structure de memoire. La memoire est mise en oeuvre sous forme d'une structure de repertoire comprenant une hierarchie en arborescence disposant de noeuds a differents niveaux de la hierarchie. Dans la structure de repertoire, des pointeurs sont d'abord ajoutes a des noeuds dont la table contient un premier nombre donne d'elements, et qui sont des noeuds comprimes en largeur. Pour maximiser le rendement de la structure de recherche fonctionnelle, l'ajout d'un pointeur a un noeud individuel comprime en largeur est permis jusqu'a ce que le nombre de pointeurs dans le noeud corresponde a une valeur definie de seuil inferieure audit premier nombre. Le noeud comprime en largeur est converti en une grappe de noeuds constituee d'un noeud pere (N50) et de differents noeuds fils (N51, ..., N54) des que le nombre de pointeurs a prendre en compte dans le noeud comprime en largeur depasse ladite valeur de seuil.

Legal Status (Type, Date, Text)
Publication 20001214 A1 With international search report.
Examination 20010208 Request for preliminary examination prior to end of 19th month from priority date

3/5/3 (Item 3 from file: 349)

00559177 **Image available**

COMPRESSION OF NODES IN A TRIE STRUCTURE
COMPRESSION DE NOEUDS DANS UNE STRUCTURE ARBORESCENTE



Patent Applicant/Assignee:
NOKIA NETWORKS OY,
IIVONEN Jukka-Pekka,
TIKKANEN Matti,
Inventor(s):
IIVONEN Jukka-Pekka ,
TIKKANEN Matti

Patent and Priority Information (Country, Number, Date):
Patent: WO 200022550 A1 20000420 (WO 0022550)
Application: WO 99FI717 19990902 (PCT/WO FI9900717)
Priority Application: FI 982095 19980929
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AE AL AM AT AU AZ BA BB BG BR BY CA CH CN CR CU CZ DE DK DM EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US UZ VN YU ZA ZW GH GM KE LS MW SD SL SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
Main International Patent Class: G06F-017/30
Publication Language: English
Fulltext Availability:
Detailed Description

Claims
Fulltext Word Count: 8244

English Abstract
The invention relates to a method for implementing a functional memory and to a memory arrangement. The memory is implemented as a directory structure comprising a tree-shaped hierarchy having nodes at several different hierarchy levels, wherein an individual node can be (i) a **trie** node associated with a logical table wherein an individual element may contain a pointer pointing to a lower node in the hierarchy, or (ii) a **bucket** containing at least one element so that the type of an individual element in the **bucket** is selected from a group including e.g. a data unit or a pointer to a stored data unit. To optimize the performance of the functional **trie** structure, the **trie** nodes are implemented as quad nodes of four elements, and in at least part of the directory structure groups of successive quad nodes are replaced by compressed nodes in such a way that (a) an individual group comprising a given quad node and its child nodes is replaced by a node whose logical table has 16 elements, and (b) a compressed node known per se is formed from said node of 16 elements by physically storing in the node only non-nil pointers and in addition a bit pattern on the basis of which the physical storage location in the node, corresponding to the search word, can be determined. The invention also relates to a structure in which no **buckets** are used.

French Abstract
L'invention concerne un procede de mise en oeuvre d'une memoire fonctionnelle et l'agencement d'une memoire. La memoire est mise en oeuvre sous la forme de structure de repertoire arborescente comportant des noeuds sur plusieurs niveaux hierarchiques differents, chaque noeud pouvant etre (i) un noeud d'arborescence associe a une table logique dans laquelle un element peut contenir un pointeur pointant vers un noeud de niveau inferieur dans la hierarchie, ou (ii) une case contenant au moins un element de maniere a selectionner le type d'element individuel dans ladite case a partir d'un groupe comprenant, par exemple, une unite de donnees ou un pointeur vers une unite de donnees en memoire. Afin d'optimiser la performance de la structure arborescente fonctionnelle, les noeuds d'arborescence sont mis en oeuvre sous la forme de noeuds quadruples de quatre elements. Dans au moins une partie de la structure de repertoire, des groupes de noeuds quadruples successifs sont remplaces par des noeuds compresses de facon (a) a remplacer chacun des groupes constitues d'un noeud quadruple donne et de ses noeuds enfants par un noeud dont la table logique a 16 elements, et (b) a former un noeud compresse a partir dudit noeud de 16 elements par mise en memoire physique dans le noeud des pointeurs non nuls seulement, et, en outre, un profil binaire sur la base duquel on peut determiner le site de mise en memoire physique dans le noeud, correspondant au mot de recherche. L'invention concerne egalement une structure dans laquelle aucune case n'est utilisee.

3/5/4 (Item 4 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00451469 **Image available**
METHOD FOR IMPLEMENTING AN ASSOCIATIVE MEMORY BASED ON A DIGITAL TRIE **STRUCTURE**
MISE EN OEUVRE D'UNE MEMOIRE ASSOCIATIVE AVEC UTILISATION D'UNE ARBORESCENCE NUMERIQUE
Patent Applicant/Assignee:
NOKIA TELECOMMUNICATIONS OY,
TIKKANEN Matti,
IIVONEN Jukka-Pekka,
Inventor(s):
TIKKANEN Matti ,
IIVONEN Jukka-Pekka
Patent and Priority Information (Country, Number, Date):

Patent: WO 9841933 A1 19980924
Application: WO 98FI192 19980304 (PCT/WO FI9800192)
Priority Application: FI 971067 19970314
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM GW HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US UZ VN YU ZW GH GM KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD TG
Main International Patent Class: G06F-017/30
Publication Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 8444

English Abstract
The invention relates to a method for implementing a memory. The memory is implemented as a directory structure comprising a tree-shaped hierachy having nodes at several different levels, wherein an individual node can be (i) a **trie** node comprising an array wherein an individual element may contain the address of a lower node in the tree-shaped hierarchy and wherein an individual element may also be empty, the number of elements in the array corresponding to a power of two, or (ii) a **bucket** containing at least one element so that the type of an individual element in the **bucket** is selected from a group including a data unit, a pointer to a stored data unit, a pointer to a node in another directory structure and another directory structure. To optimize storage space occupancy and memory efficiency, in at least part of the directory structure sets of successive **trie** nodes are replaced with compressed nodes in such a way that an individual set made up by successive **trie** nodes, from each of which there is only one address to a **trie** node at a lower level, is replaced with a compressed node (CN) storing an address to the node that the lowest node in the set to be replaced points to, information on the value of the search word by means of which said address is found, and information on the total number of bits from which search words are formed in the set to be replaced. The invention also relates to a structure in which **buckets** are not employed.

French Abstract
L'invention concerne la mise en oeuvre d'une memoire sous la forme d'une structure de repertoire arborescente comportant des noeuds sur plusieurs niveaux. Chacun des noeuds peut etre (i) un noeud d'arborescence comprenant une matrice ou un element peut soit contenir l'adresse d'un noeud inferieur dans l'arborescence, soit etre vide, le nombre d'elements de la matrice correspondant a une puissance de deux, ou (ii) une case renfermant au moins un element, de maniere a selectionner le type d'element dans la case a partir d'un groupe comprenant une unite de donnees, un pointeur designant une unite de donnees en memoire, et un pointeur designant un noeud dans une autre structure de repertoire ainsi qu'une autre structure de repertoire. Afin d'optimiser l'occupation et l'efficacite memoire, dans au moins une partie de la structure de repertoire, on remplace des ensembles de noeuds d'arborescence successifs par des noeuds compresses de facon a remplacer chacun des ensembles constitues de noeuds d'arborescence successifs, a partir desquels il n'y a qu'une seule adresse renvoyant a un noeud d'arborescence de niveau inferieur, par un noeud compresse (CN) stockant une adresse renvoyant au noeud, de facon que le noeud inferieur appartenant a l'ensemble et devant etre remplace designe, d'une part des informations relatives a la valeur du mot de recherche au moyen duquel on trouve l'adresse consideree, et d'autre part des informations relatives au nombre total de bits constituant les mots de recherche dans l'ensemble destine a etre remplace. L'invention concerne egalement une structure dans laquelle certaines cases ne sont pas utilisees.

3/5/5 **(Item 5 from file: 349)**
DIALOG(R)File 349:PCT FULLTEXT

00451468 **Image available**
METHOD FOR IMPLEMENTING AN ASSOCIATIVE MEMORY BASED ON A DIGITAL TRIE **STRUCTURE**
MISE EN OEUVRE D'UNE MEMOIRE ASSOCIATIVE SUR LA BASE D'UNE STRUCTURE NUMERIQUE ARBORESCENTE

Patent Applicant/Assignee:
NOKIA TELECOMMUNICATIONS OY,
TIKKANEN Matti,
IIVONEN Jukka-Pekka,
Inventor(s):
TIKKANEN Matti ,
IIVONEN Jukka-Pekka

Patent and Priority Information (Country, Number, Date):
Patent: WO 9841932 A1 19980924
Application: WO 98FI191 19980304 (PCT/WO FI9800191)
Priority Application: FI 971066 19970314
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM
GW HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX
NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US UZ VN YU ZW GH
GM KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH DE DK ES FI
FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD TG
Main International Patent Class: G06F-017/30
Publication Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 7736

English Abstract
The invention relates to a method for implementing a memory. The memory is implemented as a directory structure comprising a tree-shaped hierarchy having nodes at several different levels, wherein an individual node can be (i) a **trie** node comprising an array wherein an individual element may contain the address of a lower node in the tree-shaped hierarchy and wherein an individual element may also be empty, or (ii) a **bucket** containing at least one element so that the type of an individual element in the **bucket** is selected from a group including a data unit, a pointer to a stored data unit, a pointer to another directory structure and another directory structure. To minimize storage space requirement, the arrays in the **trie** nodes are implemented as quad nodes having a fixed size of four elements, and in at least part of the directory structure sets of successive quad nodes are replaced with compressed nodes in such a way that an individual set made up by successive quad nodes, from each of which there is only one address to a quad node at a lower level, is replaced with a compressed node (CN) storing an address to the quad node that the lowest node in the set to be replaced points to, information on the value of the search word by means of which said address is found, and information on the total number of bits from which search words are formed in the set to be replaced. The invention also relates to a structure in which **buckets** are not employed.

French Abstract
L'invention concerne la mise en oeuvre d'une memoire sous la forme d'une arborescence a noeuds sur plusieurs niveaux. Chacun des noeuds peut etre (i) un noeud d'arborescence comprenant une matrice ou un element peut soit contenir l'adresse d'un noeud inferieur dans l'arborescence, soit etre vide ou (ii) une case renfermant au moins un element, de maniere a selectionner le type d'element dans la case a partir d'un groupe comprenant une unite de donnees, un pointeur designant une unite de donnees en memoire, et un pointeur designant un noeud dans une autre structure de repertoire, ainsi qu'une autre structure de repertoire. Afin

de reduire les besoins en occupation memoire, les ensembles de noeuds d'arborescence sont mis en oeuvre sous forme de noeuds quadruples de taille fixe de quatre elements. Dans au moins une partie de la structure de repertoire, on remplace des ensembles des noeuds quadruples successifs par des noeuds compresses de facon a remplacer chacun des ensembles constitues de noeuds quadruples successifs a partir desquels il n'y a qu'une seule adresse renvoyant a un noeud quadruple de niveau inferieur, par un noeud compresse (CN) stockant une adresse renvoyant au noeud quadruple, de facon que le noeud inferieur appartenant a l'ensemble et devant etre remplace, designe d'une part des informations relatives a la valeur du mot de recherche au moyen duquel on trouve l'adresse consideree et d'autre part des informations relatives au nombre total de bits constituant les mots de recherche dans l'ensemble destine a etre remplace. L'invention concerne egalement une structure dans laquelle certaines cases ne sont pas utilisees.

3/5/6 (Item 6 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

00451467 **Image available**

METHOD FOR IMPLEMENTING AN ASSOCIATIVE MEMORY BASED ON A DIGITAL TRIE **STRUCTURE**

PROCEDE DE MISE EN OEUVRE D'UNE MEMOIRE ASSOCIATIVE SUR LA BASE D'UNE ARBORESCENCE NUMERIQUE

Patent Applicant/Assignee:
NOKIA TELECOMMUNICATIONS OY,
TIKKANEN Matti,
IIVONEN Jukka-Pekka,
Inventor(s):
TIKKANEN Matti ,
IIVONEN Jukka-Pekka

Patent and Priority Information (Country, Number, Date):
Patent: WO 9841931 A1 19980924
Application: WO 98FI190 19980304 (PCT/WO FI9800190)
Priority Application: FI 971065 19970314
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM
GW HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX
NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US UZ VN YU ZW GH
GM KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH DE DK ES FI
FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD TG
Main International Patent Class: G06F-017/30
Publication Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 10126

English Abstract
The invention relates to a method for implementing a memory. The memory is implemented as a directory structure comprising a tree-shaped hierarchy having nodes at several different levels, wherein an individual node can be (i) a **trie** node comprising an array wherein an individual element may contain the address of a lower node in the tree-shaped hierarchy and wherein an individual element may also be empty, the number of elements in the array corresponding to a power of two, or (ii) a **bucket** containing at least one element so that the type of an individual element in the **bucket** is selected from a group including a data unit, a pointer to a stored data unit, a pointer to another directory structure and another directory structure. To optimize storage space occupancy and memory efficiency, **trie** nodes are maintained in the directory structure in such a way that (1) in a **trie** node, the number of empty elements is smaller than or equal to half the number of elements in said node or alternatively the number of elements pointing to other **trie** nodes is

greater than a fourth of the number of elements in the node, and (2) the number of addresses in the **trie** node pointing to other **trie** nodes is smaller than or equal to half the number of elements in the node, wherein when condition (1) is faise the node is halved and when condition (2) is false the node is duplicated. The invention also relates to a structure in which **buckets** are not employed.